



COPY OF PAPER
ORIGINAL FILED

(4)

PATENT
ATTORNEY DOCKET NO. 15115.020001
PATENT APPLICATION NO. 10/091,369

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Kenichi Murai, et al. Art Unit:
Serial No.: 10/091,369 Examiner:
Filed: March 5, 2002
Title: PROGRAM UPDATE APPARATUS AND METHOD

Assistant Commissioner for Patents
Washington, DC 20231

TRANSMITTAL OF PRIORITY DOCUMENT UNDER 35 U.S.C. 119

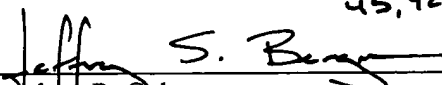
Applicant hereby confirms his claim of priority under 35 U.S.C. 119 from Japanese Application No. 2001-060598 filed March 5, 2001. A certified copy of the application from which priority is claimed is submitted herewith.

Please charge any fees due in this respect to Deposit Account No. 50-0591, referencing 15115.020001.

Respectfully submitted,

Date:

6/7/02

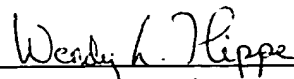
45,925

Jonathan P. Osha
Reg. No. 33,986

ROSENTHAL & OSHA L.L.P.
1221 McKinney, Suite 2800
Houston, TX 77010

Telephone: 713/228-8600
Facsimile: 713/228-8778

Date of Deposit: June 7, 2002

I hereby certify under 37 CFR 1.8(a) that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage on the date indicated above and is addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.


Wendy L. Hippe



日本国特許庁
JAPAN PATENT OFFICE

(4)

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日

Date of Application:

2001年 3月 5日

出願番号

Application Number:

特願2001-060598

[ST.10/C]:

[JP2001-060598]

出願人

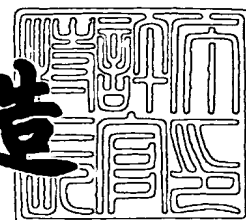
Applicant(s):

オムロン株式会社

2002年 3月29日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造



出証番号 出証特2002-3022089

【書類名】 特許願
【整理番号】 01P00370
【提出日】 平成13年 3月 5日
【あて先】 特許庁長官殿
【国際特許分類】 G06F 9/445
【発明者】

【住所又は居所】 京都府京都市下京区塩小路通堀川東入南不動堂町 8 0 1
番地 オムロン株式会社内

【氏名】 村井 謙一

【発明者】

【住所又は居所】 京都府京都市下京区塩小路通堀川東入南不動堂町 8 0 1
番地 オムロン株式会社内

【氏名】 廣野 光明

【特許出願人】

【識別番号】 000002945

【住所又は居所】 京都府京都市下京区塩小路通堀川東入南不動堂町 8 0 1
番地

【氏名又は名称】 オムロン株式会社

【代表者】 立石 義雄

【代理人】

【識別番号】 100094019

【住所又は居所】 大阪府中央区東高麗橋 4 - 3 日宝平野町ビル 4 F

【弁理士】

【氏名又は名称】 中野 雅房

【電話番号】 (06)6910-0034

【手数料の表示】

【予納台帳番号】 038508

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9800457

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 プログラム更新装置

【特許請求の範囲】

【請求項 1】 通信回線を通じて動作プログラムをアップデートするためのプログラム更新装置であって、

動作プログラムが格納されている第 1 の記憶領域とは別な第 2 の記憶領域を確保する手段と、

第 1 の記憶領域に格納されている第 1 の動作プログラムとは別な動作プログラムを第 2 の記憶領域に格納する手段と、

第 1 の記憶領域に格納されている動作プログラムと第 2 の記憶領域に格納されている動作プログラムのうち、いずれかを選択的に起動させる手段とを備えたことを特徴とするプログラム更新装置。

【請求項 2】 メモリ機器により動作プログラムをアップデートするためのプログラム更新装置であって、

動作プログラムが格納されている第 1 の記憶領域とは別な第 2 の記憶領域を確保する手段と、

第 1 の記憶領域に格納されている第 1 の動作プログラムとは別な動作プログラムを第 2 の記憶領域に格納する手段と、

第 1 の記憶領域に格納されている動作プログラムと第 2 の記憶領域に格納されている動作プログラムのうち、いずれかを選択的に起動させる手段とを備えたことを特徴とするプログラム更新装置。

【請求項 3】 動作プログラムの起動の失敗を検知する手段を備え、

前記第 1 及び第 2 の記憶領域のうち一方の記憶領域に格納された動作プログラムが起動に失敗した場合には、他方の記憶領域に格納されている動作プログラムを自動的に起動させるようにした、請求項 1 又は 2 に記載のプログラム更新装置。

【請求項 4】 前記第 1 又は第 2 の記憶領域に格納されている動作プログラム上で動作するアプリケーションプログラムが起動に失敗した場合には、修復アプリケーションにより当該アプリケーションプログラムを自動的に修復させる

ようにした、請求項 1 又は 2 に記載のプログラム更新装置。

【請求項 5】 前記動作プログラムは、オペレーティングシステムを含むものである、請求項 1 又は 2 に記載のプログラム更新装置。

【請求項 6】 前記動作プログラムは、J a v a プログラム実行環境であることを特徴とする、請求項 1 又は 2 に記載のプログラム更新装置。

【請求項 7】 前記動作プログラムのチェックをチェックサムを用いて行なうようにした、請求項 1 又は 2 に記載のプログラム更新装置。

【請求項 8】 前記動作プログラムの異常検出をウォッチドックタイマのタイムアウトを用いて行なうようにした、請求項 1 又は 2 に記載のプログラム更新装置。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、プログラム更新装置に関する。特に、本発明は、機器組込みシステムのバグフィックスやバージョンアップを行なう際に用いられるプログラム更新装置に関するものである。

【0 0 0 2】

【従来の技術】

機器組込みシステム等のプログラムでは、バグを解消するためのバグフィックスや、プログラムの改良、データの更新などに伴うバージョンアップが必要となることがある。そのため、従来からも機器組込みシステム等のプログラムの書き換えが行なわれることがある。

【0 0 0 3】

図 1 はエネルギー使用量を計測する家庭用の使用量計測器の場合を表している。使用量計測器 1 は家屋の外壁等に設置されており、家庭 2 で使用される機器 3 には、第 1 のエネルギー供給ライン 4 や第 2 のエネルギー供給ライン 5 等を通じ、使用量計測器 1 を経由してエネルギーが供給されている。例えば、第 1 のエネルギー供給ライン 4 を通じて供給されるエネルギーとは電気のことであり、使用量計測器 1 はその電気使用量を計測し、積算している。また、第 2 のエネルギー

供給ライン 5 を通じて供給されるエネルギーとはガスのことであり、使用量計測器 1 はそのガス使用量を計測し、積算している。

【 0 0 0 4 】

このような使用量計測器の目的は、電気やガス等のエネルギーの使用量を計測することであるが、これらのデータは料金徴収の目的以外にも使用されている。例えば、エネルギー供給元にとっては、1 日ないし 1 週間の各時間ごとの細かい消費傾向や季節ごとの消費傾向を知ることができれば、そのデータに基づき、エネルギー消費者に対して各時間毎の細かな料金設定を行なうことが可能になり、サービスの向上を図ることができる。また、使用ピークの時間帯には高いエネルギー使用料金を設定することで、エネルギーの総消費量を抑制し、エネルギー消費が特定の時間帯に集中してエネルギーの供給不足になるのを防止することができる。

【 0 0 0 5 】

上記のように料金徴収やデータ活用のためには、エネルギー供給元は、各家庭の使用量計測器を検針してエネルギー使用量を調べる必要がある。そのため、図 2 に示すように、管理センタ 6（電気会社やガス会社等のエネルギー供給元）から派遣される計測人（検針人）7 が各家庭 2 を訪問し、使用量計測器 1 のメータを検針している。しかし、このようにして人手作業によって各家庭 2 の使用量計測器 1 を検針するという方法では、1 日に検針できる戸数が限られており、人件費もかさみ、非常に効率が悪かった。許容可能な効率を達成しようとすれば、15 分などの短時間で 1 軒の検針を行なわなければならない、人手作業では、ほぼ達成不可能であった。

【 0 0 0 6 】

このようなことから、家庭に設置された使用量計測器を遠隔地から自動検針できるようにしたものがある。このような従来例では、図 3 に示すように、電柱 8 の上などにデータ収集器 9 を設置しておき、データ収集器 9 を中継して各家庭 2 の使用量計測器 1 を管理センタ 6 につないでいる。データ収集器 9 とは、一定時間毎に各家庭 2 の使用量計測器 1 からデータを収集し、そのデータを管理センタ 6 へ送る機能を持った機器である。データ収集器 9 は、電源の供給に困らない、

電波的に遮断物が少ない、人に触れて欲しくないといったことから、電柱 8 の上など人の行きにくいところに設置されている。各家庭 2 の使用量計測器 1 や管理センタ 6 とデータ収集器 9 との間は、通信回線 10、11 などによって結ばれる。

【0007】

このような方式によれば、多くの家庭の使用量計測器を遠隔地（管理センタ 6）から瞬時に自動点検することができ、人件費の削減を図れ、また 1 日に検針できる件数も制限が無くなり、効率的に検針を行なうことができるようになっている。

【0008】

また、別な例として、家庭内に設置されている機器を外部から操作する場合を考える。図 4 に示す例では、家庭 2 内に設置されている機器 12A、12B、…と機器制御器 14 とを信号線 13 を通じて結んでおき、電話回線やネットワーク等の通信回線 15 に接続された機器制御器 14 を経由して各機器 12A、12B、…を家庭 2 の外から操作したり、機器 12A、12B、…からの情報を外部で取得できるようにしている。例えば、機器 12A が監視カメラであれば、電話回線に接続したモバイルコンピュータや携帯電話により外部から監視カメラの画像をモニターしたり、監視カメラの撮影方向を操作したりすることができ、機器 12B がビデオレコーダであれば、ビデオレコーダの録画予約を外部から行なったり、機器 12C がホームセキュリティに必要な監視センサであれば、遠隔地から外部にいてセキュリティシステムからの通知を取得することができる。

【0009】

機器制御器 14 に接続される機器にはさまざまなものが想定され、機器ごとに持っている情報が違うため、機器制御器 14 に機器を接続するにあたっては、その機器に応じて機器制御器 14 のシステムプログラムを書き換える必要がある。

【0010】

また、ホームセキュリティに関連するシステムで用いられている場合には、ホームセキュリティ動作中では機器制御器 14 のだんまり（フリーズ）は許されない。ホームセキュリティとは、不審者の侵入、火災、ガス漏れ、非常通報、緊急

・救急の検知をするものであり、機器制御器 14 がだんまりになる（フリーズする）とは、機器の応答が無くなり、外部へ状態を通知できなくなったり、外部からの入力を受け付けない状態になることであるから、機器制御器 14 がだんまり状態になると、ホームセキュリティが意味をなさず、機器制御器 14 の信頼性が損なわれるからである。

【0011】

【発明が解決しようとする課題】

図 5 は通信回線を通じて外部と結ばれた機器組込みシステム、例えば上記のようなデータ収集器や機器制御器における技術的課題を説明する図解図である。ここで、解決状況とは、データ収集器 9 や機器制御器 14 において発生する解決課題を表している。対応策とは、この解決課題を解決する手段を表している。目的とは、最終的に達成したい事柄を表している。前提条件とは、この目的を達成する際に問題となる前提条件を表している。必要性・必然性とは、この前提条件から導かれる必要条件を表している。また、必達課題とは、上記目的を達成するために要求される解決課題を表している。以下、図 5 に沿って説明する。

【0012】

データ収集器 9 や機器制御器 14 においては、システム（プログラム）にバグが見つかることがあり（ステップ S 1）、そのような場合にはシステムプログラムのバグフィックスが必要になる（ステップ S 2）。また、データ収集器 9 や機器制御器 14 に新しい機器（例えば、使用量計測器 1 など）を追加接続した場合（ステップ S 3）や、データ収集器 9 や機器制御器 14 に新しい機能を追加したい場合（ステップ S 4）には、それぞれのシステムプログラムにも対応する機能を追加したり、システムプログラムをアップデート（更新）したりする必要がある（ステップ S 5）。そして、バグフィックスや機能追加のためにプログラムをアップデートするためには、データ収集器 9 や機器制御器 14 に組み込まれているシステムのプログラムを書き換える必要がある（ステップ S 6）。

【0013】

このようなプログラムの書き換えが必要となる場合、例えばアップデート用のプログラムを提供してユーザーサイドで書き換えをして貰うということになると

、機械に苦手なユーザーではプログラムの書き換えやフリーズした際の復旧作業を行なって貰うことは期待できない（ステップS7）。また、個々のデータ収集器9や機器制御器14はプログラム提供者（管理センタなど）から遠隔地にあることが多く、しかも、データ収集器9や機器制御器14では人の行きにくい場所（電柱の上など）に設置されていることが多い（ステップS8）。よって、システムプログラムの書き換えを行なう場合には、人の手を介さず、通信回線等を通じて、自動的に書き換えを行なう必要がある（ステップS9）。

【0014】

さらに、ホームセキュリティに関連した機器制御器14等では、書き換えのためにシステムを停止させることはできず（ステップS10）、システムを稼働させた状況下で書き換えを行なう必要があり（ステップS11）、書き換え中にシステムがフリーズしたような場合でも、とりあえず再起動させてシステムを稼働させる必要がある（ステップS12）。

【0015】

よって、機器組込みシステムによっては、通信回線等を通じて自動的にプログラムの書き換えを行えると共に万一の場合にもシステムがフリーズすることのないアップデート方法が求められている（ステップS13）。

【0016】

従来でも、通信回線を通じて機器組込みシステムのプログラムを書き換える処理は行なわれている。しかし、それは、正しく動くプログラムが前提で、それが正しく書き込めたか否かをチェックするだけであった。そのため、致命的な部分は書き換えないようにしている。また、プログラムのチェックにチェックサムのみを利用している場合には、複数のエラーが発生した場合、チェックする値が偶然正しい値に一致する可能性があり、正しく書き込めたと誤認識する恐れがあった。

【0017】

さらに、従来の書き換え方式では、バグのあるプログラムには対応できず、オペレーティングシステムが動作しなければ、ネットワークダウンロードなど複雑な処理は行えなかった。

【 0 0 1 8 】

【発明の開示】

本発明の目的とするところは、機器組込みシステム（特に、常時動作する必要のある機器のシステム）において、自動的にシステムのアップデートを行え、しかもシステムをフリーズさせることのないプログラム更新装置を提供することにある。

【 0 0 1 9 】

本発明にかかるプログラム更新装置は、通信回線を通じて動作プログラムをアップデートするためのプログラム更新装置であって、動作プログラムが格納されている第1の記憶領域とは別な第2の記憶領域を確保する手段と、第1の記憶領域に格納されている第1の動作プログラムとは別な動作プログラムを第2の記憶領域に格納する手段と、第1の記憶領域に格納されている動作プログラムと第2の記憶領域に格納されている動作プログラムのうち、いずれかを選択的に起動させる手段とを備えたことを特徴としている。ここで、動作プログラムとは、機器とアプリケーションを制御するプログラムであって、オペレーティングシステム（OS）も含まれる。また、通信回線は、有線でも無線でもよく、例えばインターネットや携帯電話、PHSによるネットワーク（通信網）を用いることができる。

【 0 0 2 0 】

このプログラム更新装置にあっては、第1及び第2の記憶領域のうち一方の記憶領域にそれまで動作していた動作プログラムを残したまま、他方の記憶領域に新しい動作プログラムをダウンロードし、第1の記憶領域に格納されている動作プログラムと第2の記憶領域に格納されている動作プログラムのうち、いずれかを選択的に起動させることができるので、いずれか一方の記憶領域に新しい動作プログラムをダウンロードした後、その動作プログラムの起動に失敗した場合には、他方の記憶領域に保存されている以前の動作プログラムを再起動させることがき、動作プログラムのアップデートに失敗しても動作プログラムを組み込まれている機器がフリーズするのを回避できる。また、このプログラム更新装置にあっては、通信回線を介して遠隔から自動的にアップデートを行なわせることが

できる。従って、このプログラム更新装置によれば、遠隔から自動的に動作プログラムをアップデートさせることができ、しかもアップデートに失敗した場合でもシステムがフリーズするのを防止することができる。

【 0 0 2 1 】

また、本発明の別なプログラム更新装置は、メモリ機器により動作プログラムをアップデートするためのプログラム更新装置であって、動作プログラムが格納されている第1の記憶領域とは別な第2の記憶領域を確保する手段と、第1の記憶領域に格納されている第1の動作プログラムとは別な動作プログラムを第2の記憶領域に格納する手段と、第1の記憶領域に格納されている動作プログラムと第2の記憶領域に格納されている動作プログラムのうち、いずれかを選択的に起動させる手段とを備えたことを特徴としている。ここで、メモリ機器とは、プログラム更新装置に接続することのできるメモリであれば特に限定されるものでなく、メモ리카ード、メモリスティックなどがある。

【 0 0 2 2 】

このプログラム更新装置にあっても、第1及び第2の記憶領域のうち一方の記憶領域にそれまで動作していた動作プログラムを残したままで、他方の記憶領域に新しい動作プログラムをロードさせ、第1の記憶領域に格納されている動作プログラムと第2の記憶領域に格納されている動作プログラムのうち、いずれかを選択的に起動させることができるので、いずれか一方の記憶領域に新しい動作プログラムをロードした後、その動作プログラムの起動に失敗した場合には、他方の記憶領域に保存されている以前の動作プログラムを再起動させることができ、動作プログラムのアップデートに失敗しても動作プログラムを組み込まれている機器がフリーズするのを回避できる。従って、このプログラム更新装置によれば、自動的に動作プログラムをアップデートさせることができ、しかもアップデートに失敗した場合でもシステムがフリーズするのを防止することができる。

【 0 0 2 3 】

また、上記各プログラム更新装置においては、動作プログラムの起動の失敗を検知する手段を備え、前記第1及び第2の記憶領域のうち一方の記憶領域に格納された動作プログラムが起動に失敗した場合には、他方の記憶領域に格納されて

いる動作プログラムを自動的に起動させるようにするのが望ましい。このような実施態様によれば、新しい動作プログラムの起動に失敗した場合には、自動的に古い動作プログラムが起動されるので、自動的にアップデートを行なわせる場合でも、システムのフリーズを回避することができる。

【 0 0 2 4 】

また、この実施形態において、動作プログラムの起動失敗を検知する手段としては、チェックサムを用いて動作プログラムをチェックするものでもよく、ウォッチドックタイマのタイムアウトにより動作プログラムをチェックするものでもよい。

【 0 0 2 5 】

また、上記各プログラム更新装置においては、第 1 又は第 2 の記憶領域に格納されている動作プログラムの上で動作するアプリケーションプログラムが起動に失敗した場合には、修復アプリケーションにより当該アプリケーションプログラムを自動的に修復させるようにするのが望ましい。ここで、アプリケーションプログラムとは、動作プログラム上で動く、機器の利用目的を実行するプログラムである。また、修復アプリケーションとは、動作プログラム上で動く、修復機能を持ったプログラムである。このような実施態様によれば、いずれかの動作プログラムが起動に成功した後（すなわち、ダウンロードした動作プログラムが起動に成功した後、あるいはダウンロードした動作プログラムが起動に失敗して古い動作プログラムが再起動された後）、アプリケーションプログラムが起動に失敗した場合には、修復プログラムを自動的に働かせてアプリケーションプログラムを修復させることができるので、より安全にアップデート作業を行なうことができる。

【 0 0 2 6 】

また、本発明のプログラム更新装置は、J a v a プログラム実行環境下において用いることもできる。

【 0 0 2 7 】

なお、この発明の以上説明した構成要素は、可能な限り組み合わせることができる。

【 0 0 2 8 】

【発明の実施の形態】

本発明に係るプログラム更新方式は、図6に示すように、ダウンロードステップ（ステップS22）、新起動ステップ（ステップS22）、復旧起動ステップ（ステップS23）、アップデート作業の完了（ステップS24）からなる。そして、図7に示すように、システムのアップデートが開始すると（ステップS21）、システムに新動作プログラム（アップデートされた動作プログラム）をダウンロードし、システム内のプログラム動作とは別領域にある記憶装置空き領域（例えば、複数のROM領域において、旧動作プログラムが稼働中でないROM領域）に新動作プログラムを転送し、書き込みを行なう（ダウンロードステップ、ステップS22）。新動作プログラムがダウンロードされて転送と書き込みが終了したら、ダウンロードの時点で動作していた旧動作プログラムに代えてダウンロードした新動作プログラムを起動させる（新起動ステップ、ステップS23）。そして、新動作プログラムが正常に起動したか否かを判定し（ステップS24）、新動作プログラムが正常に起動していたら、正常解決機能によってアップデートが完了したと判断し、アップデート作業を完了する（ステップS26）。

【 0 0 2 9 】

これに対し、ステップS24において、新動作プログラムが正常に起動していないと判断された場合には、復旧挑戦機能および確実復旧機能を内容とするシステム復旧動作を行なってシステムを再起動させてフリーズから復旧させ（復旧起動ステップ、ステップS25）、アップデート作業を完了する（ステップS26）。

【 0 0 3 0 】

ここで上記正常解決機能とは、ダウンロードした新動作プログラムが正常起動し、システムの動作プログラムが旧動作プログラムから新動作プログラムに切り換わってシステムが新動作プログラムで動作するようにする機能であり、復旧挑戦機能および確実復旧機能とは、ダウンロードした新動作プログラム等が正常起動できない場合に、システムがフリーズしないよう復旧させるための機能である。

【 0 0 3 1 】

しかして、本発明に係るプログラム更新方式では、旧動作プログラムが稼働しているシステムに新動作プログラムをダウンロードし、稼働するプログラムを旧動作プログラムから新動作プログラムに切り替えて新動作プログラムを実行させる。そして、フリーズ等により新動作プログラムの起動に失敗した場合には、もとの旧動作プログラムを再起動し、旧プログラムを実行させる。あるいは、旧動作プログラム又は新動作プログラムが稼働している状態で、実行アプリケーションがフリーズ等により実行停止した場合には、修復アプリケーションを実行して実行アプリケーションを修復させるものである。ここで、動作プログラムとは機器とアプリケーションを制御するプログラムであって、オペレーティングシステム（OS）も含まれる。また、旧動作プログラムとは、ダウンロード実行時に稼働中の動作プログラムであり、新動作プログラムとは、置換用の動作プログラム（アップデートプログラム）である。実行アプリケーションとは、動作プログラム上で動く、機器の利用目的を実行するアプリケーションプログラムである。修復アプリケーションとは、動作プログラム上で動く、修復機能を持った必要最低限のアプリケーションプログラムである。このアップデート作業においては、遠隔からプログラムのアップデートを行なうため、通信回線を通じてプログラムをダウンロードする。通信回線は、有線でも無線でもよく、例えばインターネットや携帯電話、PHSによるネットワーク（通信網）を用いることができる。

【 0 0 3 2 】

なお、システムのフリーズは、(1)ダウンロードした動作プログラムが破壊されている場合、(2)初期化データなどのデータが破壊されている場合、あるいは、(3)動作プログラムは正常に動作していても実行アプリケーションが正常に起動しない場合に起きるが、本発明のアップデート方式では、以下に説明するようにしてシステムのフリーズ（だんまり）を回避する。

【 0 0 3 3 】

図 8 は上記方式を具体的に説明する図であって、図 9 はその処理順序を図示したフロー図である。図 8（a）はダウンロード前の状態を表しており、システム内では旧動作プログラム 2 1 により実行アプリケーション 2 2 が動作している。

なお、図8において、符号「RUN」は動作中のアプリケーションを表している。ついで、アップデート作業が開始すると、図8(b)に示すように、ネットワーク等の通信回線を通じてシステムにデータ25がダウンロードされ、図8(c)に示すように新動作プログラム24が空き領域（例えば、複数のROM領域ROM1とROM2があり、ROM1の領域が稼働中であるとすれば、ROM2の領域）に転送され、そこに書き込まれる。新動作プログラム24の書き込みが終了すると、図8(d)に示すように、正常解決機能によって書き込まれた領域（例えば、上記ROM2の領域）から新動作プログラム24が起動され、動作プログラムが旧動作プログラム21から新動作プログラム24に切り替えられ、実行アプリケーション22が新動作プログラム24の上で実行され、修復アプリケーション23も新動作プログラム24の上で実行可能となる。

【0034】

新動作プログラム24を起動すると（新起動ステップ、ステップS31）、新動作プログラム24の起動に成功（OK）したか、失敗（NG）したか判断し、新動作プログラム24のダウンロードが正常完了していて新動作プログラム24の起動に成功していれば（図8(e)、ステップS32）、アップデート機能が完了する（図8(f)、ステップS33）。

【0035】

このようにして正常解決機能が働いてダウンロードした新動作プログラム24でシステムが正常に起動されると、図10に示すように、機器の目的を妨げることなく（すなわち、システムを停止させることなく）システムの書き換えが完了し、機器の目的を実行しつつ、同時にシステムの書き換えを行なうという本発明の目的が達成される。

【0036】

これに対し、新動作プログラム24の起動に失敗した場合には（ステップS34）、失敗の状況に応じて復旧挑戦機能、第1又は第2の确实復旧機能の3種類のシステム復旧機能のうちから最適な方法を実行する。すなわち、復旧挑戦機能とは、正常解決機能による新動作プログラム24の起動に失敗した場合に、図8(g)に示すように、ダウンロードした新動作プログラム24を実行せず、元の

旧動作プログラム 21 を再起動して旧動作プログラム 21 でシステムを実行し直すものである。また、第 1 の確実復旧機能とは、正常解決機能により新動作プログラム 24 が正常に起動されても、実行アプリケーション 22 の起動に失敗した場合、図 8 (h) に示すように、ダウンロードした新動作プログラム 24 が起動された状況において、修復アプリケーション 23 を実行して実行アプリケーション 22 を修復するものである。第 2 の確実復旧機能とは、正常解決機能による新動作プログラム 24 の起動に失敗した後、復旧挑戦機能によって旧動作プログラム 21 が復旧されたが、実行アプリケーション 22 の起動に失敗した場合に、図 8 (i) に示すように、旧動作プログラム 21 が実行されている状況において、修復アプリケーション 23 を実行して実行アプリケーション 22 を修復するものである。

【0037】

新動作プログラム 24 の起動失敗の状況が、新動作プログラム 24 は正常に起動したが、実行アプリケーション 22 がフリーズとなったことである場合には（ステップ S 35）、第 1 の確実復旧機能を働かせて実行アプリケーション 22 を修復し（図 8 (h) (j)、ステップ S 36）、新動作プログラム 24 の下で実行アプリケーション 22 を動作させ、アップデート機能が完了する（図 8 (f)、ステップ S 33）。また、新動作プログラム 24 がフリーズした場合には（ステップ S 37）、復旧挑戦機能を働かせて動作プログラムを新動作プログラム 24 から旧動作プログラム 21 に戻して旧動作プログラム 21 を再起動し（図 8 (g)、ステップ S 38）、システムが正常に動くようになれば修復を完了し（図 8 (j)）、旧動作プログラム 21 の下で実行アプリケーション 22 を動作させ、アップデート機能が完了する（図 8 (f)、ステップ S 33）。もし、復旧挑戦機能を作動させてもシステムが復旧されなかった場合には（ステップ S 39）、すなわち旧動作プログラム 21 は正常に再起動したが、実行アプリケーション 22 がフリーズした場合には、第 2 の確実復旧機能を働かせて旧動作プログラム 21 の下で修復アプリケーション 23 を実行して実行アプリケーション 22 を修復し（図 8 (i) (j)、ステップ S 40）、旧動作プログラム 21 の下で実行アプリケーション 22 を動作させ、アップデート機能が完了する（図 8 (f)、

ステップS33)。

【0038】

このようにして正常解決機能が失敗した場合でも、復旧挑戦機能によって旧動作プログラムでシステムを再起動させることができ、あるいは确实復旧機能によって修復アプリケーションで実行アプリケーションを修復することができるので、決してシステムがだんまり状態（フリーズ）に陥ることが無く、新動作プログラムの起動に失敗してもシステムをフリーズさせないという本発明の目的が達成される。

【0039】

次に、上記のようなりモートプログラム更新方式を実行するためのシステム構成を図12に示す。このシステムはデータ収集器や機器制御器等の機器に組み込まれており、マイクロプロセッサ（CPU）26、パワーオンリセット（Power On Reset）回路27、ウォッチドックタイマ（WDT）28、ブートステータスレジスタ29、ブートプログラムROM30、プログラムROM31、32、RAM33、入出力部（I/O）34、ネットワーク35、メモリカード36等によって構成され、これらはバスライン37を通じて結ばれている。ここで、マイクロプロセッサ26は、システム全体を制御するものである。パワーオンリセット回路27は、システムをオン、オフし、あるいはハードウェアリセットを掛けるための回路である。ウォッチドックタイマ28は、システムがタイムアウトした際にソフトウェアリセットを掛けることができるものである。ブートプログラムROM30は、最小限のブートプログラム領域である。プログラムROM31、32は、2重化されたプログラム格納領域である。ブートプログラムROM30及びプログラムROM31、32は、同一メモリの領域を分割したものであってもよい。RAM33は、正常解決機能等を実行する際など、一時的にデータを保存するものである。ネットワーク35は、インターネット等を通じて外部（公衆回線等）と結ばれており、新動作プログラムはネットワーク35を通じていずれか一方のプログラムROM31、32にダウンロードされる。また、新動作プログラム24は、入出力部34やメモリカード36からインストールできるようにしてもよい。

【0040】

ブートステータスレジスタ29は、起動させた動作プログラムや実行アプリケーションが異常となったかどうかを記憶しておくレジスタであって、ソフトウェアリセットによっては値が変更されない。ブートステータスレジスタ29は、図13に示すように、最下位ビットにBOOTフラグを持ち、その上位ビットにCONFIGフラグを持っている。BOOTフラグは、電源オンの直後には”0”となっているが、初回ブートイメージ異常があると値”1”が保存される。また、CONFIGフラグは、電源オンの直後には”0”となっているが、初期設定値に異常があると値”1”が保存される。システムは、新動作プログラム24を起動した際、新動作プログラム24の異常でソフトウェアリセットがかかると、ブートステータスレジスタ29のBOOTフラグをチェックし、旧動作プログラム21が格納されているROM領域と新動作プログラム24が格納されているROM領域のうちいずれのROM領域から動作プログラムを起動するかを決定する。また、実行アプリケーション22の異常でソフトウェアリセットがかかると、システムは、ブートステータスレジスタ29のCONFIGフラグをチェックし、実行アプリケーション22と修復アプリケーション23のうちどちらのアプリケーションを起動するか決定する。

【0041】

図14は図12に示したシステムを機能として表現したものである。これらの機能は、

- (f1) ダウンロードされた新動作プログラム24の格納場所（メモリ領域）を決定する機能、
- (f2) 決定された格納場所へダウンロードされた新動作プログラム24を格納する機能、
- (f3) 格納された新動作プログラム24をチェックする機能、
- (f4) 格納された新動作プログラム24を起動する機能、
- (f5) ダウンロードする前に動作していた旧動作プログラム21を起動する機能、
- (f6) 起動された新動作プログラム24または旧動作プログラム21の動作

をチェックする機能、

(f 7) チェックされた動作プログラム上で実行アプリケーション 22 を起動する機能、

(f 8) チェックされた動作プログラム上で修復アプリケーション 23 を起動する機能、

(f 9) 起動された実行アプリケーション 22 または修復アプリケーション 23 の動作をチェックする機能、

から構成されており、新動作プログラム 24 のダウンロードから始まって図 13 の矢印の順に実行され、アップデート機能の完了によって終了する。

【0042】

しかして、図 12～図 14 のようなシステム構成において、本発明のリモートプログラム更新方式は図 15 に示す手順に沿って実行される。以下、図 14 の機能を参照しながら、図 15 に沿ってアップデートの手順を説明する。新動作プログラム 24 がネットワーク 35 を通じて外部からダウンロードされ、いずれかの ROM に格納されると (ステップ S41、機能 f1、f2)、システムは新動作プログラムが正常に起動したかどうかチェックする (ステップ S42)。すなわち、新動作プログラム 24 がダウンロードされると、システムは、格納された新動作プログラム 24 をチェックし (機能 f3)、異常がなければ新動作プログラム 24 を起動し (機能 f4)、さらに、ブートステータスレジスタ 29 をチェックすることにより、起動された新動作プログラム 24 が正常に起動したかどうかチェックする (機能 f6)。

【0043】

新動作プログラム 24 が正常に起動している場合には、起動された新動作プログラム 24 上で実行アプリケーション 22 を起動させ (機能 f7)、実行アプリケーションが正常に起動したかどうかチェックする (ステップ S43)。すなわち、システムは、ブートステータスレジスタ 29 をチェックすることによって実行アプリケーションに異常がないかチェックする (機能 f9)。そして、実行アプリケーションが正常に起動していれば、システムの起動を完了する (アップデート完了、ステップ S45)。

【0044】

これに対し、ステップS43で実行アプリケーション22に異常があった場合には、ブートステータスレジスタ29のCONFIGフラグが”1”になっているので、ウォッチドックタイマ28のオーバータイムによってソフトウェアリセットが掛かり、ブートステータスレジスタ29のCONFIGフラグがチェックされると、修復アプリケーション23が起動される。起動された修復アプリケーション23はシステム（実行アプリケーション22）を修復し（ステップS44、機能f8）、システムの起動を完了する（ステップS45）。

【0045】

また、ステップS42で新動作プログラム24が異常である場合には、ブートステータスレジスタ29のBOOTフラグが”1”になっているので、ウォッチドックタイマ28のオーバータイムによってソフトウェアリセットが掛かり、ブートステータスレジスタ29のBOOTフラグがチェックされると、旧動作プログラム21でシステムを再起動してシステムを旧動作プログラム21に復旧させる（ステップS46、機能f5）。ついで、旧動作プログラム21上で実行アプリケーション22を起動させ（機能f7）、実行アプリケーション22が正常に起動したかどうかチェックする（ステップS47）。そして、実行アプリケーション22が正常に起動していれば、システムの起動を完了する（アップデート完了、ステップS45）。これに対し、実行アプリケーション22が異常である場合には、ウォッチドックタイマ28のオーバータイムによってソフトウェアリセットが掛かり、ブートステータスレジスタ29のCONFIGフラグがチェックされると、修復アプリケーション23を起動してシステム（実行アプリケーション22）を修復し（ステップS48、機能f8）、システムの起動を完了する（ステップS45）。

【0046】

次に、OS（オペレーティングシステム）上でJavaVM（仮想マシン）が動いているJava実行環境下においてアップデートが行なわれる様子を具体的に説明する。図16はOS上でJavaVMが動いているJava実行環境の一例を示しており、システム起動時に働くブートプログラム41、システム全体を

制御するOS 42、OS 42の違いを吸収して共通のプラットフォームを提供するJava VM 43、Java VM上で動くクラスライブラリ 44、機器の利用目的を実行する実行アプリケーション 46、必要最小限の修復機能を持った修復アプリケーション 47によって構成されている。ここで、OS 42、Java VM 43及びクラスライブラリ 44がアップデート対象となる動作プログラム 45である。

【0047】

一方、上記動作プログラム 45をアップデートするためのアップデート実行部は、新しいJava VM及びOSと以前からのJava VM及びOSのうちいずれの動作プログラムを起動させるかを決める起動プログラム決定部 48、どのアプリケーションプログラムを起動させるかを決定する起動アプリケーション決定部 49、起動したアプリケーションの動作を監視するアプリケーション動作監視部 50を備えている。

【0048】

この環境下におけるアップデート方式でも、正常解決機能、復旧挑戦機能、第1及び第2の确实復旧機能が実現されているが、まず、これらの機能の詳細は省き、動作プログラムをダウンロードしてシステムをアップグレードする際の全体の流れを図17及び図18により説明する。このシステムでは、新しいJava VM及びOS（新動作プログラム）をネットワーク 35等を通じて外部からダウンロードすると（ステップS51）、ダウンロードされたJava VM及びOSは、現在動作しているJava VM及びOS（旧動作プログラム）により使用されていないROM領域に格納される（ステップS52）。ついで、システムにハードウェアリセットまたはソフトウェアリセットを掛け、システムを再起動する（ステップS53）。その際、システムの起動プログラム決定部 48は、ブートステータスレジスタ 29をチェックすることにより、新旧いずれのJava VM及びOSを起動させるかを決定する（ステップS54）。起動するJava VM及びOSが決まると、システムは決定されたJava VM及びOSが正常か否かをチェックサムを用いてチェックする（ステップS55）。このチェックにより、決定されたJava VM及びOSが正常でないと判断されると、起動するJa

Java VM及びOSを他方のJava VM及びOSに変更する（ステップS56）

【0049】

こうして正常に起動できるJava VM及びOSが決定されると、ウォッチドックタイマ28が起動し（ステップS57）、ブートステータスレジスタ29のBOOTフラグを反転させ（ステップS58）、Java VM及びOSを起動する（ステップS59）。

【0050】

ついで、起動アプリケーション決定部49が、起動させるアプリケーションを決定すると（ステップS60）、決定したアプリケーションをロードし（ステップS61）、ブートステータスレジスタ29のBOOTフラグとCONFIGフラグを反転させる（S62）。これによって、ステップS58で反転されたBOOTフラグが元に戻る。また、この段階に達するまでにJava VM及びOSがフリーズしていれば、ウォッチドックタイマ28のタイムアウトでソフトウェアリセットがかかっている筈であるから、この時点で、Java VM及びOSが正常に起動したと判断できる。従って、ウォッチドックタイマ28を更新し（ステップS63）、アプリケーション動作監視部50により、ロードしたアプリケーションの動作を監視する（ステップS64）。

【0051】

図19～図24は、図17及び図18に示した処理中に行なわれる正常解決機能、復旧挑戦機能、确实復旧機能の処理手順を示している。なお、図19～図24において、共通部とは、正常解決機能、復旧挑戦機能、确实復旧機能に共通な図17、18の処理を指している。まず、図19により正常解決機能における起動プログラム決定部48の処理を説明する。図17のステップS58でブートステータスレジスタ29のBOOTフラグが反転された後、起動プログラム決定部48はブートステータスレジスタ29のBOOTフラグをチェックする（ステップS71）。そして、起動プログラム決定部48は、BOOTフラグが反転しているか否か調べ（ステップS72）、反転していれば、ダウンロードする前のJava VM+OSで起動するように決定し（ステップS74）、BOOTフラグ

が反転していなければダウンロードした J a v a VM+OS で起動するように決定する（ステップ S 7 3）。ダウンロードした J a v a VM 及び OS に異常がなく正常解決機能が働く場合には、BOOT フラグは反転していないので、起動プログラム決定部 4 8 はダウンロードした J a v a VM 及び OS を起動するように決定する（ステップ S 7 3）。

【 0 0 5 2 】

また、図 2 0 は復旧挑戦機能における起動プログラム決定部 4 8 の処理を説明している。図 1 7 のステップ S 5 8 でブートステータスレジスタ 2 9 の BOOT フラグが反転された後、起動プログラム決定部 4 8 はブートステータスレジスタ 2 9 の BOOT フラグをチェックする（ステップ S 7 . 1）。そして、起動プログラム決定部 4 8 は、BOOT フラグが反転しているか否か調べ（ステップ S 7 2）、BOOT フラグが反転していれば、ダウンロードする前の J a v a VM+OS で起動するように決定し（ステップ S 7 4）、BOOT フラグが反転していなければダウンロードした J a v a VM+OS で起動するように決定する（ステップ S 7 3）。ダウンロードした J a v a VM 及び OS に異常があつて復旧挑戦機能が働く場合には、正常解決機能で動作中に、図 1 8 のステップ S 6 2 に達する前にウォッチドックタイマ 2 8 がタイムアウトしてソフトウェアリセットがかかり、BOOT フラグは反転したままになるので、起動プログラム決定部 4 8 はダウンロードする前の J a v a VM 及び OS を起動するように決定する（ステップ S 7 4）。

【 0 0 5 3 】

また、図 2 1 は确实復旧機能における起動プログラム決定部 4 8 の処理を説明している。図 1 7 のステップ S 5 8 でブートステータスレジスタ 2 9 の BOOT フラグが反転された後、起動プログラム決定部 4 8 はブートステータスレジスタ 2 9 の BOOT フラグをチェックする（ステップ S 7 5）。そして、起動プログラム決定部 4 8 は、BOOT フラグが反転しているか否か調べ（ステップ S 7 6）、BOOT フラグが反転していれば、直前に動作していない J a v a VM+OS を起動するように決定し（ステップ S 7 8）、BOOT フラグが反転していなければ直前に動作していた J a v a VM+OS で起動するように決定する（ステ

ップS77)。确实復旧機能が働く場合には、第1の确实復旧機能でも第2の确实復旧機能でもJava VM及びOSの正常起動が確認されているので、ブートステータスレジスタ29のBOOTフラグは反転しておらず、起動プログラム決定部48は直前に動作していたJava VM及びOS（第1の确实復旧機能の場合には、ダウンロードされたJava VM及びOS、第2の确实復旧機能の場合には、ダウンロードする前のJava VM及びOS）を起動するように決定する（ステップS74）。

【0054】

次に、図22により正常解決機能及び復旧挑戦機能における起動アプリケーション決定部49の処理を説明する。図18のステップS62でブートステータスレジスタ29のCONFIGフラグが反転された後、起動アプリケーション決定部49はブートステータスレジスタ29のCONFIGフラグをチェックする（ステップS79）。そして、起動アプリケーション決定部49は、CONFIGフラグが反転しているか否か調べ（ステップS80）、反転していれば、修復アプリケーションを起動するようにCONFIGファイルを書き換え（ステップS81）、システムのCONFIGを初期化してプログラムを実行し（ステップS82）、ブートステータスレジスタ29のCONFIGフラグを反転させた（ステップS83）後、システムをソフトウェアリセットさせる（ステップS84）。ステップS80でCONFIGフラグが反転していなければ、起動アプリケーション決定部49は、CONFIGフラグで設定されているアプリケーションを起動するように決定する（ステップS85）。正常解決機能、復旧挑戦機能が動作している場合には、CONFIGフラグは反転していないので、起動アプリケーション決定部49は、CONFIGフラグで設定されているアプリケーションを起動するように決定する（ステップS85）。

【0055】

また、図23は确实復旧機能における起動アプリケーション決定部49の処理を説明している。図18のステップS62でブートステータスレジスタ29のCONFIGフラグが反転された後、起動アプリケーション決定部49はブートステータスレジスタ29のCONFIGフラグをチェックする（ステップS79）

。そして、起動アプリケーション決定部49は、CONFIGフラグが反転しているか否か調べ（ステップS80）、反転していれば、修復アプリケーションを起動するようにCONFIGファイルを書き換え（ステップS81）、CONFIG初期化プログラムを実行し（ステップS82）、ブートステータスレジスタ29のCONFIGフラグを反転させた（ステップS83）後、システムをソフトウェアリセットさせる（ステップS84）。ステップS80でCONFIGフラグが反転していなければ、起動アプリケーション決定部49は、CONFIGフラグで設定されているアプリケーションを起動するように決定する（ステップS85）。第1又は第2の确实復旧機能が動作している場合には、正常解決機能、復旧挑戦機能でJava VM+OSの正常動作が確認されているが、アプリケーション動作監視部50でウォッチドックタイマ28がタイムアウトしたため、CONFIGフラグが反転しているので、起動アプリケーション決定部49は、修復アプリケーション23を起動するようにCONFIGファイルを書き換え（ステップS81）、CONFIGフラグ初期化プログラムを実行し（ステップS82）、ブートステータスレジスタ29のCONFIGフラグを反転させる（ステップS83）。これにより、図17のステップS58で反転されたCONFIGフラグはもとに戻る。さらに、起動アプリケーション決定部49は、システムにソフトウェアリセットを掛ける（ステップS84）。ソフトウェアリセットを掛けて、再度このステップへ来た時には、CONFIGフラグはもとに戻っているので、CONFIGファイルで設定されているアプリケーション（修復アプリケーション23）を起動するように決定される。

【0056】

次に、図24により正常解決機能、復旧挑戦機能、确实復旧機能におけるアプリケーション動作監視部50の処理を説明する。図18のステップS63でウォッチドックタイマ28がリセットされた（ステップS86）後、アプリケーション動作監視部50は通常処理を行ない（ステップS87）、修復アプリケーション23が起動しているか否かチェックする（ステップS88）。修復アプリケーション23が起動していない場合には、ウォッチドックタイマ28がタイムアウトしているかチェックするが、修復アプリケーション23が起動している場合に

は、ウォッチドックタイマ 2 8 をチェックせず、ステップ S 8 6 の処理へ戻ってウォッチドックタイマ 2 8 をリセットする。従って、确实復旧機能が実行されている場合には、修復アプリケーション 2 3 が起動しているので、決してウォッチドックタイマ 2 8 がタイムアウトしない。これに対し、実行アプリケーション 2 2 が実行されている正常解決機能または復旧挑戦機能の場合には、アプリケーション動作監視部 5 0 はウォッチドックタイマ 2 8 のタイムアウトをチェックし（ステップ S 8 9）、ウォッチドックタイマ 2 8 がタイムアウトならソフトウェアリセットを掛けて（ステップ S 9 0）确实復旧機能へ移行し、タイムアウトしていなければ、ステップ S 8 6 の処理へ戻ってウォッチドックタイマ 2 8 をリセットする。

【 0 0 5 7 】

しかして、ダウンロードした J a v a VM + O S （新動作プログラム）が正常に起動し、実行アプリケーション 2 2 も正常に起動した場合（正常解決機能の場合）の処理は、図 1 9、図 2 2、図 2 4 の流れになる。また、ダウンロードした J a v a VM + O S （新動作プログラム）が正常に起動せず、ダウンロードする前の J a v a VM + O S （旧動作プログラム）で正常に復旧した場合（復旧挑戦機能の場合）の処理は、図 2 0、図 2 2、図 2 4 の流れになる。また、正常解決機能において、ダウンロードした J a v a VM + O S （新動作プログラム）は正常に起動したが、実行アプリケーション 2 2 が正常に動作せず、修復アプリケーション 2 3 で確実に復旧した場合（第 1 の确实復旧機能の場合）の処理は、図 2 1、図 2 3、図 2 4 の流れになる。また、确实復旧機能において、ダウンロードする前の J a v a VM + O S （旧動作プログラム）で実行アプリケーション 2 2 が正常に動作せず、修復アプリケーション 2 3 で確実に復旧した場合（第 2 の确实復旧機能の場合）の処理は、図 2 1、図 2 3、図 2 4 の流れになる。そして、正常解決機能、復旧挑戦機能、确实復旧機能の 3 段階を段階的に行なうことで、本発明の目的が実現される。

【 0 0 5 8 】

つきに、正常解決機能におけるソフトウェアの流れを図 2 5 に示す。この図に示すように、正常解決機能の場合には、ダウンロード部 5 1 によって新しい J a

v a 及びOS（新動作プログラム24）がダウンロードされ、現在動作していないJ a v a VM及びOS（旧動作プログラム21）を新しいJ a v a VM及びOSに置き換える。ついで、ブートプログラムを起動する。J a v a VM+OS起動／動作監視部52は、起動するJ a v a VM及びOSを決定し、壊れていないか、フリーズしないかをチェックし、フリーズすれば、復旧挑戦機能を実行する。また、アプリケーション起動／動作監視部53は、実行アプリケーション22を起動し、フリーズしないかチェックし、フリーズしたらソフトウェアリセットを掛け、第1の确实復旧機能を実行する。

【0059】

また、復旧挑戦機能におけるソフトウェアの流れを図26に示す。新しいJ a v a VM及びOSの起動時にフリーズし、ソフトウェアリセットが掛かると、J a v a VM+OS起動／動作監視部52は、ダウンロードする前に動作していた古いJ a v a VM及びOS（旧動作プログラム）を起動し直す。ついで、アプリケーション起動／動作監視部53は、実行アプリケーション22を起動し、フリーズしないかチェックする。そして、フリーズしたら、ソフトウェアリセットを掛け、第2の确实復旧機能を実行する。

【0060】

また、第1の确实復旧機能におけるソフトウェアの流れを図27に示す。アプリケーション起動／動作監視部53は実行アプリケーション22を監視しており、新しいJ a v a VM及びOSで実行アプリケーション動作時にフリーズし、ソフトウェアリセットがかかると、新しいJ a v a VM及びOSは正常に起動することを確認しているので、J a v a VM+OS起動／動作監視部52により新しいJ a v a VM及びOSを起動し直す。ついで、アプリケーション起動／動作監視部53が修復アプリケーションを起動し、復旧が完了する。

【0061】

また、第2の确实復旧機能におけるソフトウェアの流れを図28に示す。アプリケーション起動／動作監視部53は実行アプリケーション22を監視しており、古いJ a v a VM及びOSで実行アプリケーションが動作時にフリーズし、ソフトウェアリセットがかかると、古いJ a v a VM及びOSは正常に起動するこ

とを確認されているので、Java VM+OS起動/動作監視部52により古いJava VM及びOSを起動し直す。ついで、アプリケーション起動/動作監視部53は、修復アプリケーション23を起動し、復旧を完了する。

【0062】

図29は自動検針システム（データ収集器）61のモジュール構成図である。この自動検針システム61は、PHS62を介して家庭や管理センタと定期的にデータ通信を行なっている。そして、図29に示すアップデート可能部分63が管理センタからのデータによってアップデートされるようになっている。図30は、図29の自動検針システム61のハードウェア構成図である。PHS62はPPPドライバやシリアルドライバを介してシリアルポート64と接続されている。ROM65には、起動プログラムやブートプログラム、CONFIGファイルが格納される。また、外部記憶装置であるコンパクトフラッシュメモリ（CF）66には、自動検針用アプリケーション70、修復アプリケーション71、自動検針により得られたデータなどが格納される。図31は、図30のROM65とコンパクトフラッシュメモリ66の状態を示している。図31に示すROM65のROM領域67で、起動プログラムMが立ち上がり、自動検針用アプリケーションが動作している。図29のアップデート可能部分63において、バグフィックスを行なったり、ソフト的な機能追加やハード（機器）の追加によるドライバの追加を行なった場合には、起動プログラムMの対応する個所を修正し、コンパイルする。修正、コンパイル済みの起動プログラムをM' とすると、起動プログラムM' をネットワークを通じて、図31のROM65のROM領域68へ格納し、ROM領域68から起動し直す。また、起動プログラムM' の起動に失敗した場合には、ROM領域67のMで起動し直す。また、CONFIGファイル69のデータの破壊により自動検針用アプリケーション70の起動に失敗した場合には、図31のコンパクトフラッシュメモリ66に格納している修復アプリケーション71でデータの修復を行ない、必要最低限の機能を持ったアプリケーションで起動し直す。これにより、自動でアップデートを行え、かつ決してフリーズしないアップデートが完了する。

【0063】

【発明の効果】

本発明のプログラム更新装置によれば、システムのアップデート作業において、自動的にアップデートを行なうことができ、かつ、新しい動作プログラム等が起動に失敗しても決してシステムがフリーズすること無く、アップデートを行なうことができる。

【図面の簡単な説明】

【図 1】

エネルギー使用量を計測する家庭用の使用量計測器の使用状態を示す図である。

【図 2】

家庭に設置された使用量計測器を検針する方法を説明する図である。

【図 3】

家庭に設置された使用量計測器を遠隔地から自動検針できるようにした構成を説明する図である。

【図 4】

家庭内に設置されている機器を外部から操作するための機器制御器を説明する図である。

【図 5】

通信回線を通じて外部と結ばれた、例えばデータ収集器や機器制御器等の機器組込みシステムにおける技術的課題を説明する図解図である。

【図 6】

本発明に係るプログラム更新方式を説明する概略図である。

【図 7】

図 6 のプログラム更新方式の実行手順を示すフロー図である。

【図 8】

図 6 のプログラム更新方式を具体的に説明する図である。

【図 9】

図 8 に対応させて上記プログラム更新方式の実行手順を示す図である。

【図 1 0】

正常解決機能と本発明の目的との関係を表した図である。

【図 1 1】

復旧挑戦機能および確実復旧機能と本発明の目的との関係を表した図である。

【図 1 2】

本発明にかかるプログラム更新方式を実行するためのシステム構成を示す図である。

【図 1 3】

ブートステータスレジスタの構造を示す図である。

【図 1 4】

図 1 2 に示したシステムを機能として表現した図である。

【図 1 5】

図 1 2 ～図 1 4 のシステム構成におけるアップデート方法の手順を示す図である。

【図 1 6】

OS 上で J a v a V M が動いている J a v a 実行環境の一例を示す図である。

【図 1 7】

動作プログラムをダウンロードしてシステムをアップグレードする際の全体の流れを表した図である。

【図 1 8】

図 1 7 の続図である。

【図 1 9】

正常解決機能における起動プログラム決定部の処理を説明する図である。

【図 2 0】

復旧挑戦機能における起動プログラム決定部の処理を説明する図である。

【図 2 1】

確実復旧機能における起動プログラム決定部の処理を説明する図である。

【図 2 2】

正常解決機能及び復旧挑戦機能における起動アプリケーション決定部の処理を説明する図である。

【図 2 3】

确实復旧機能における起動アプリケーション決定部の処理を説明する図である。

【図 2 4】

正常解決機能、復旧挑戦機能、确实復旧機能におけるアプリケーション動作監視部の処理を説明する図である。

【図 2 5】

正常解決機能におけるソフトウェアの流れを示すフロー図である。

【図 2 6】

復旧挑戦機能におけるソフトウェアの流れを示す図である。

【図 2 7】

第 1 の确实復旧機能におけるソフトウェアの流れを示す図である。

【図 2 8】

第 2 の确实復旧機能におけるソフトウェアの流れを示す図である。

【図 2 9】

自動検針システム（データ収集器）のモジュール構成図である。

【図 3 0】

図 2 9 の自動検針システムのハードウェア構成図である。

【図 3 1】

図 3 0 の ROM とコンパクトフラッシュメモリの状態を示す図である。

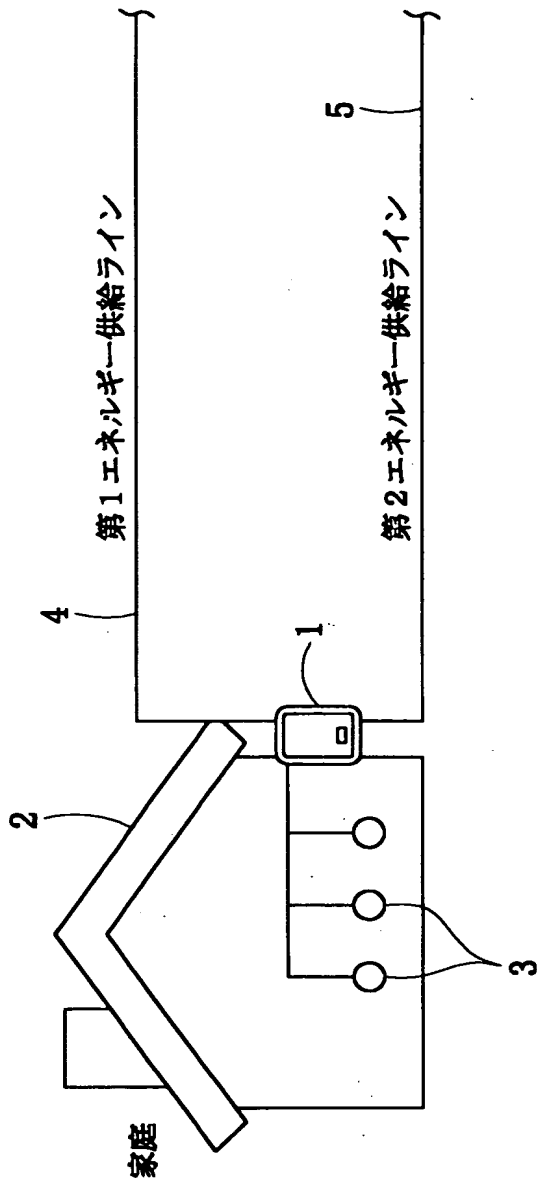
【符号の説明】

- 2 1 旧動作プログラム
- 2 2 実行アプリケーション
- 2 3 修復アプリケーション
- 2 4 新動作プログラム
- 2 6 マイクロプロセッサ
- 2 7 パワーオンリセット回路
- 2 8 ウォッチドックタイマ
- 2 9 ブートステータスレジスタ

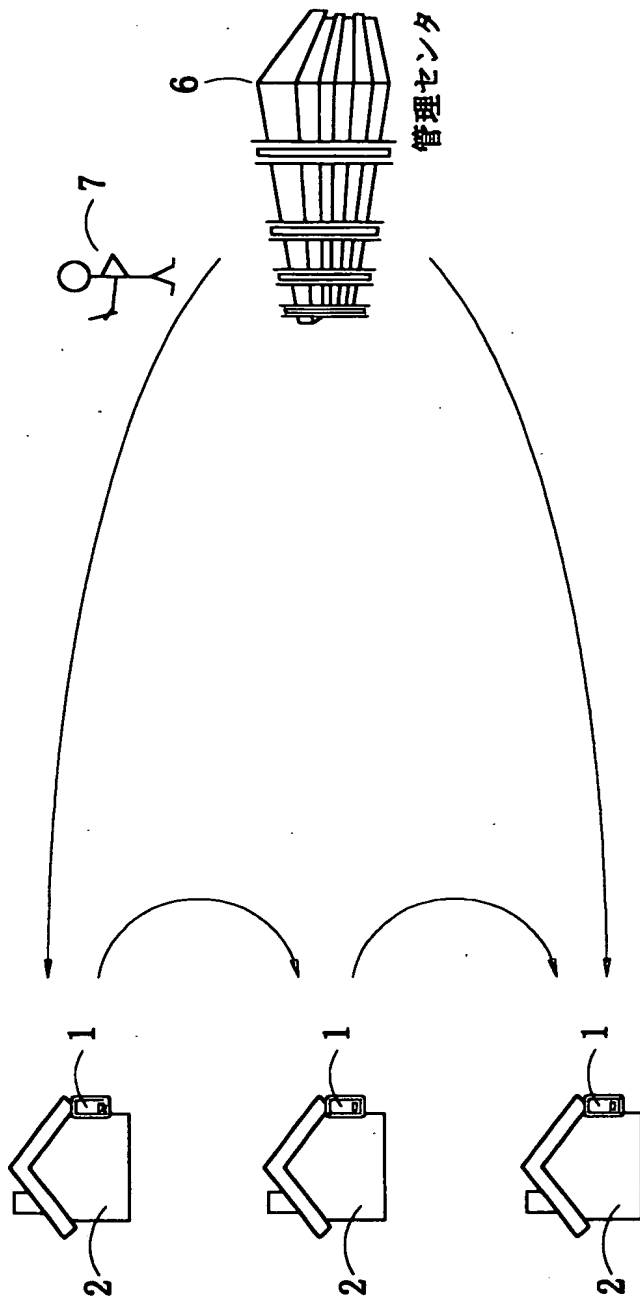
- 30 ブートプログラムROM
- 31、32 プログラムROM
- 35 ネットワーク
- 36 メモリカード
- 41 ブートプログラム
- 42 OS
- 43 JavaVM
- 44 クラスライブラリ
- 45 動作プログラム
- 46 実行アプリケーション
- 47 修復アプリケーション

【書類名】 図面

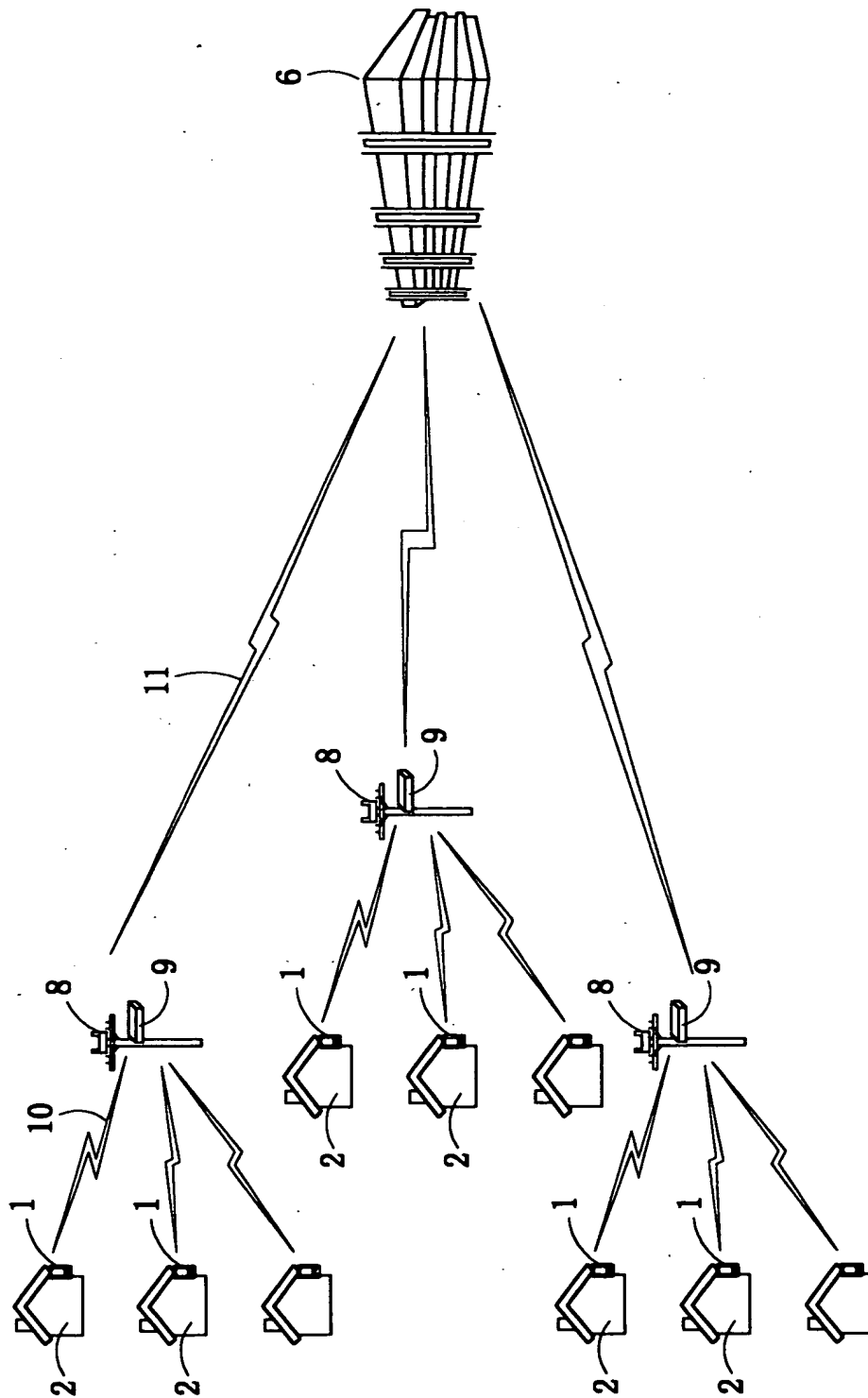
【図 1】



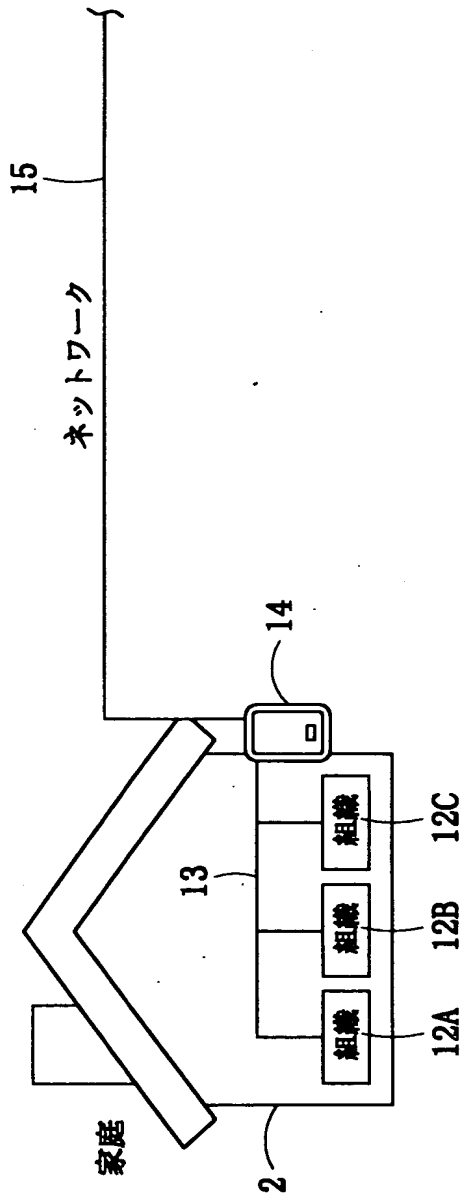
【図 2】



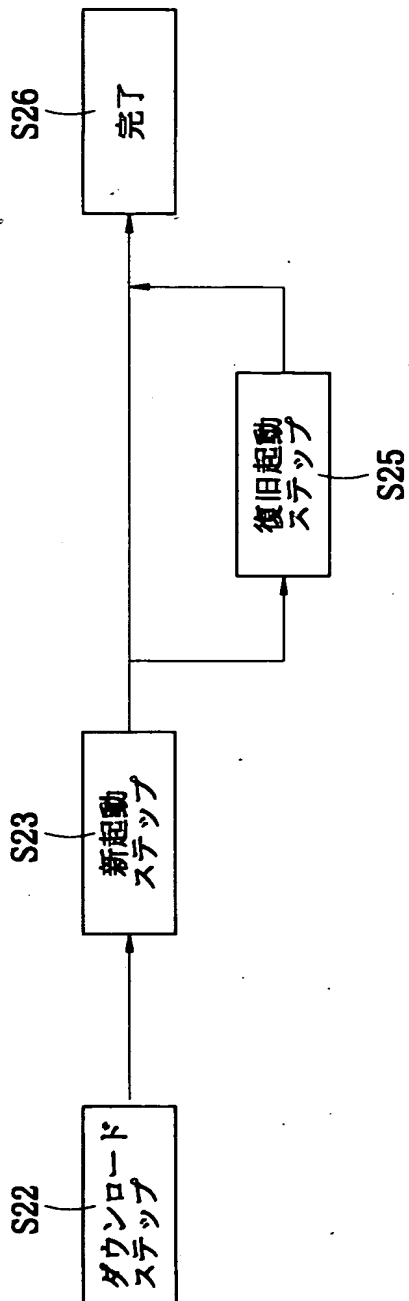
【図 3】



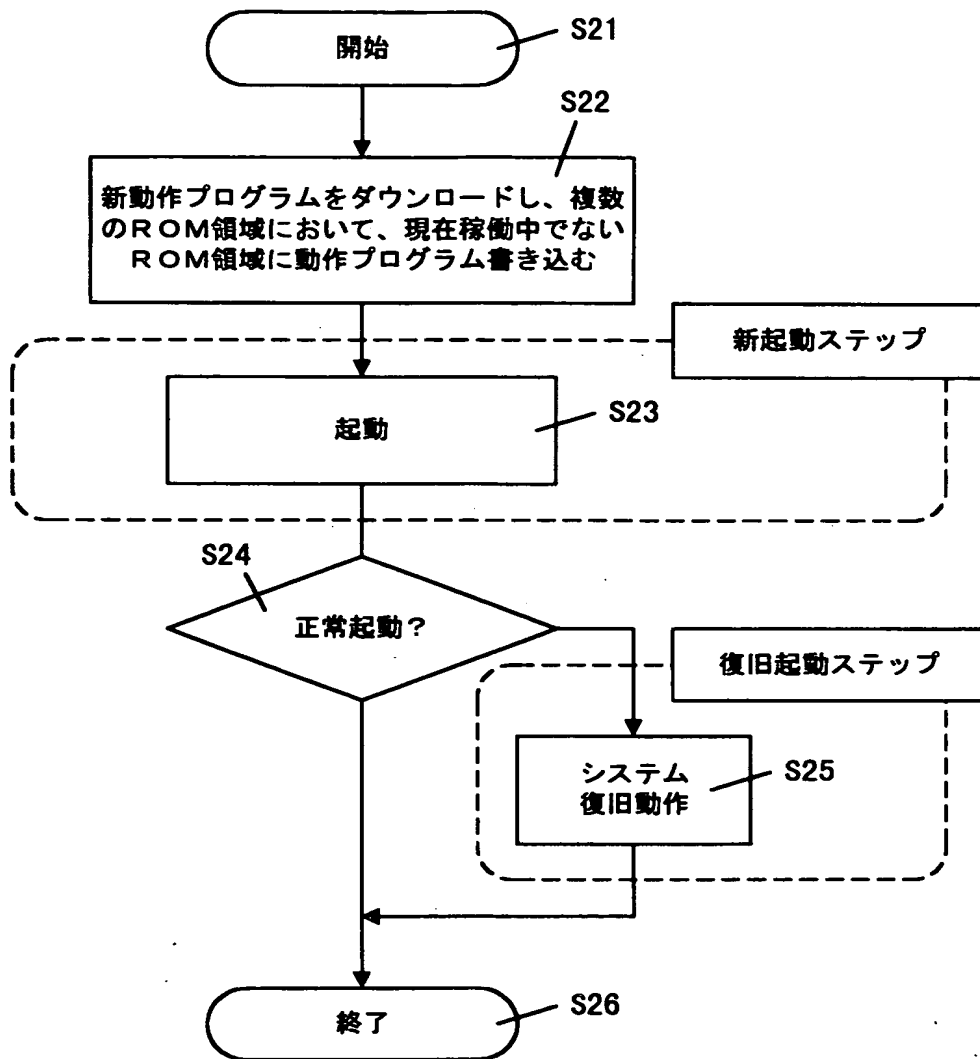
【図 4】



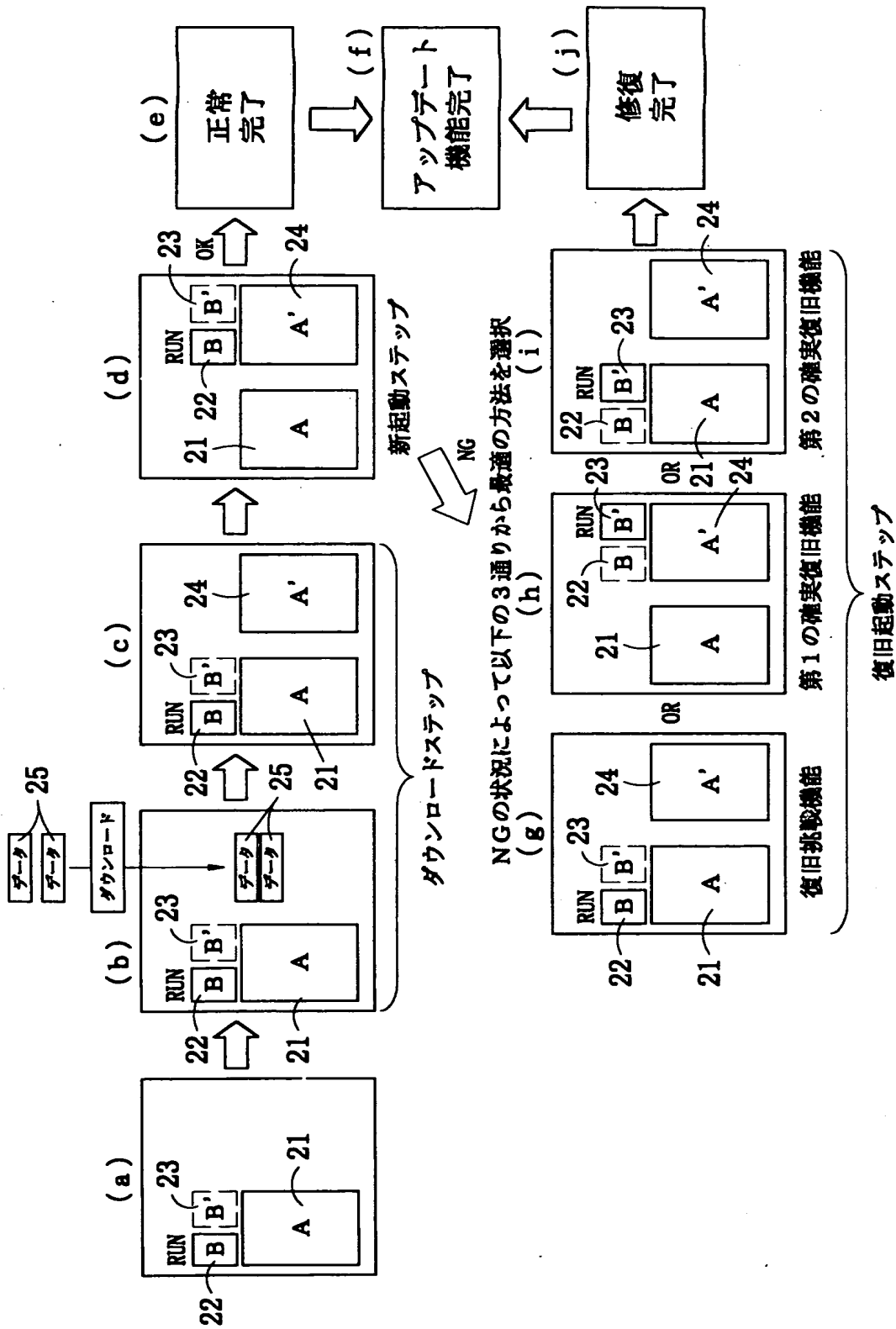
【図6】



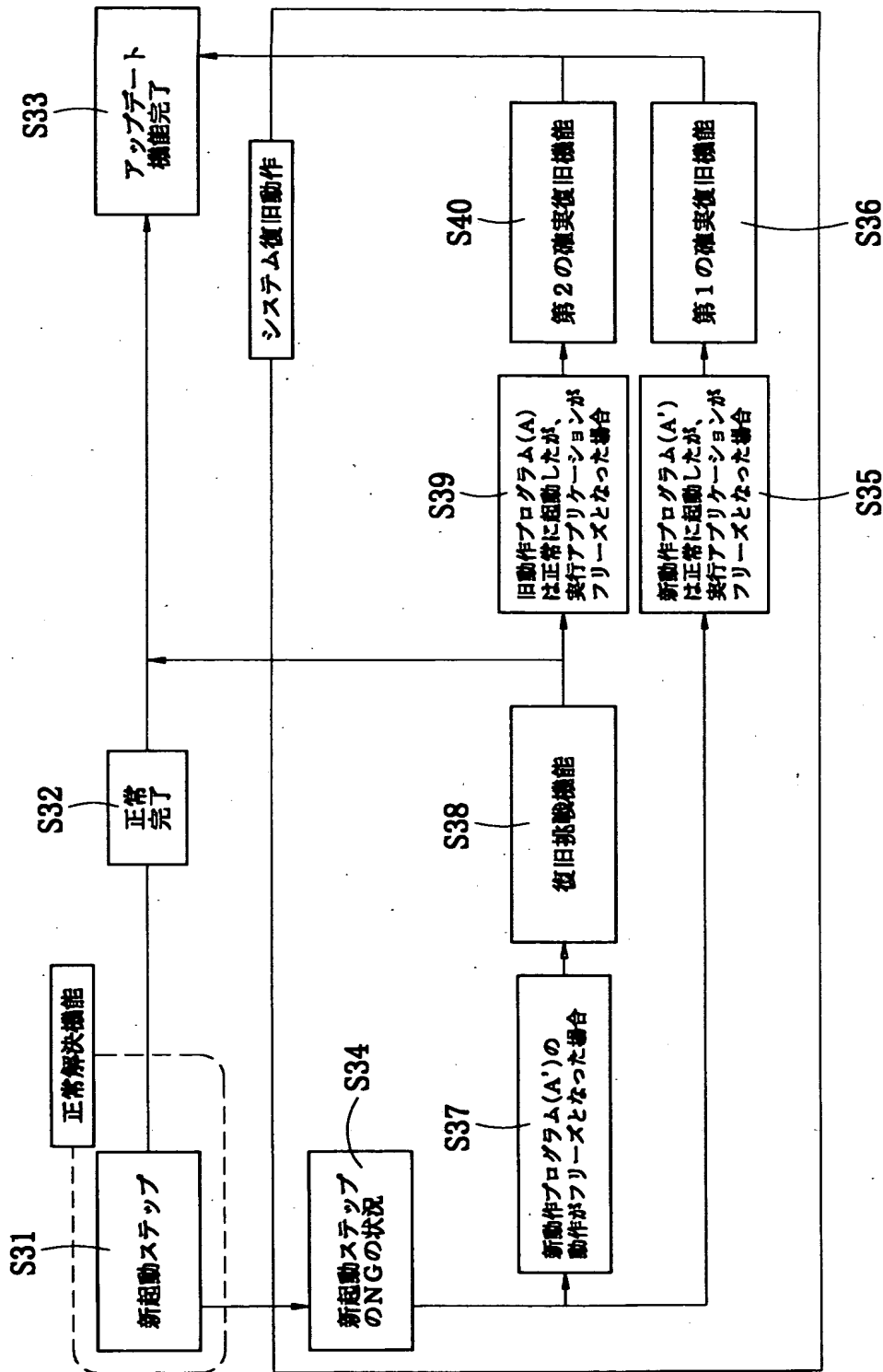
【図 7】



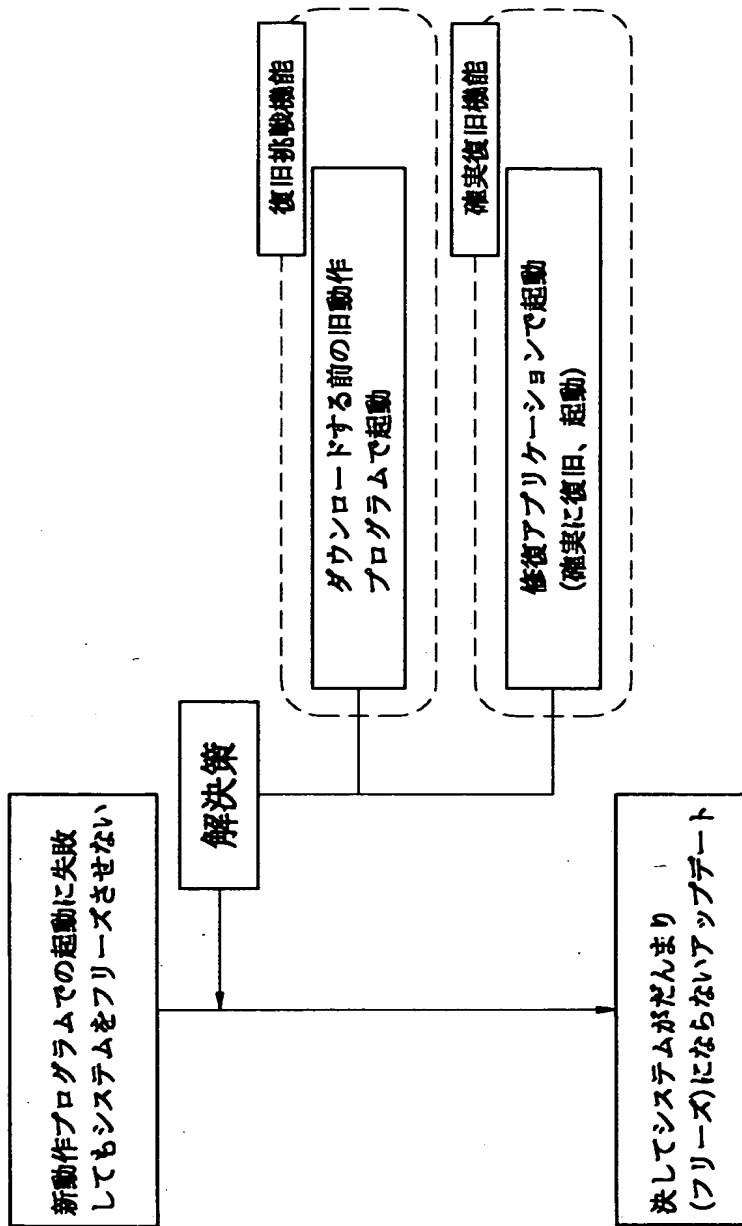
【図 8】



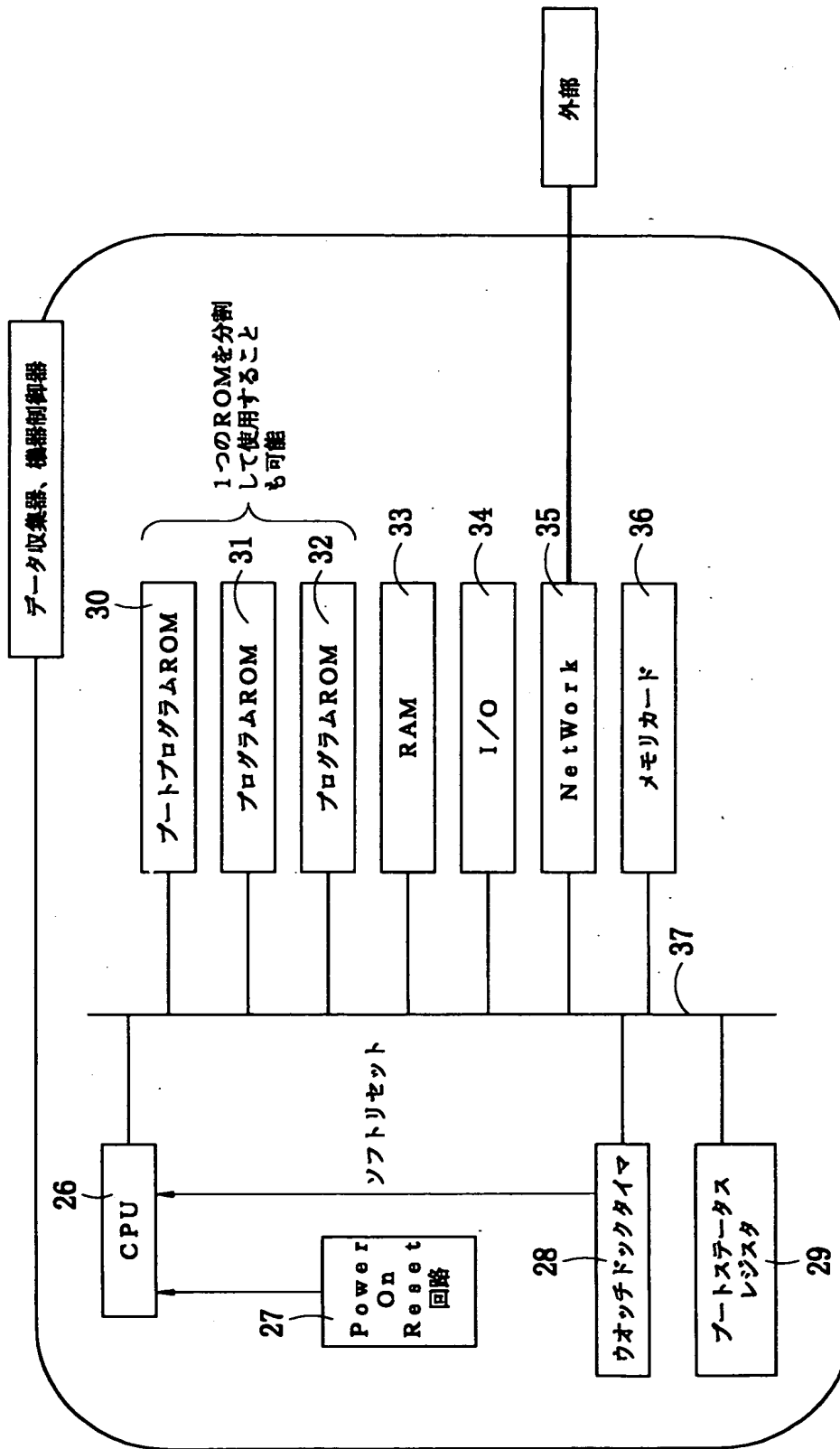
【図9】



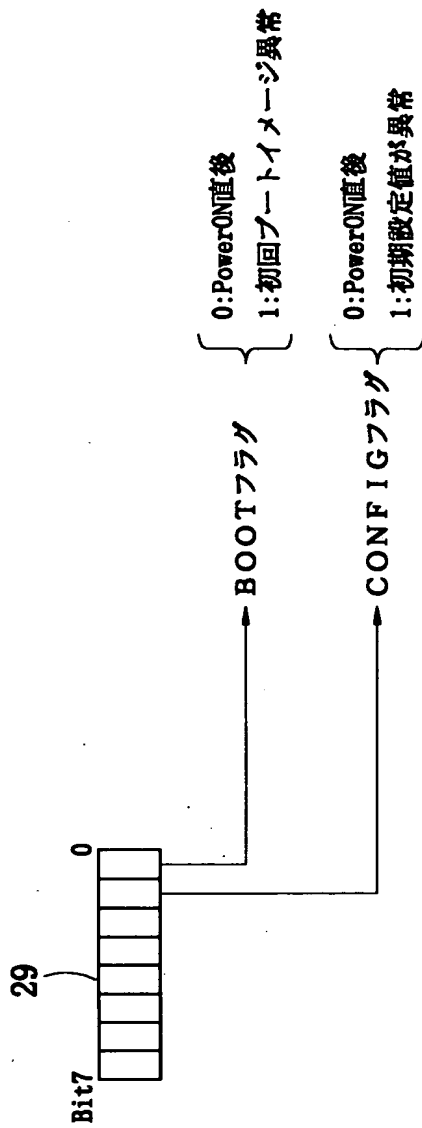
【図 11】



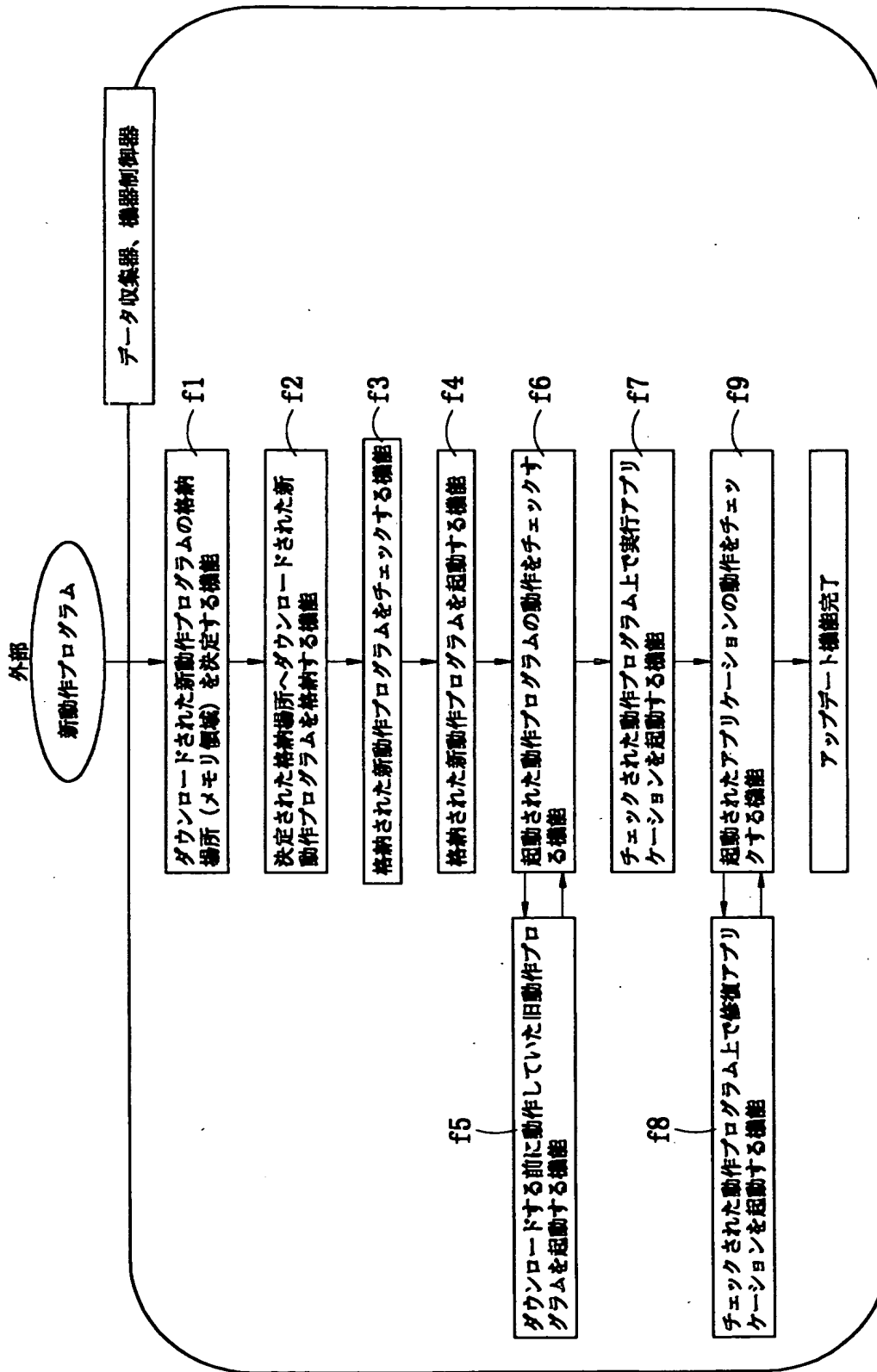
【図 12】



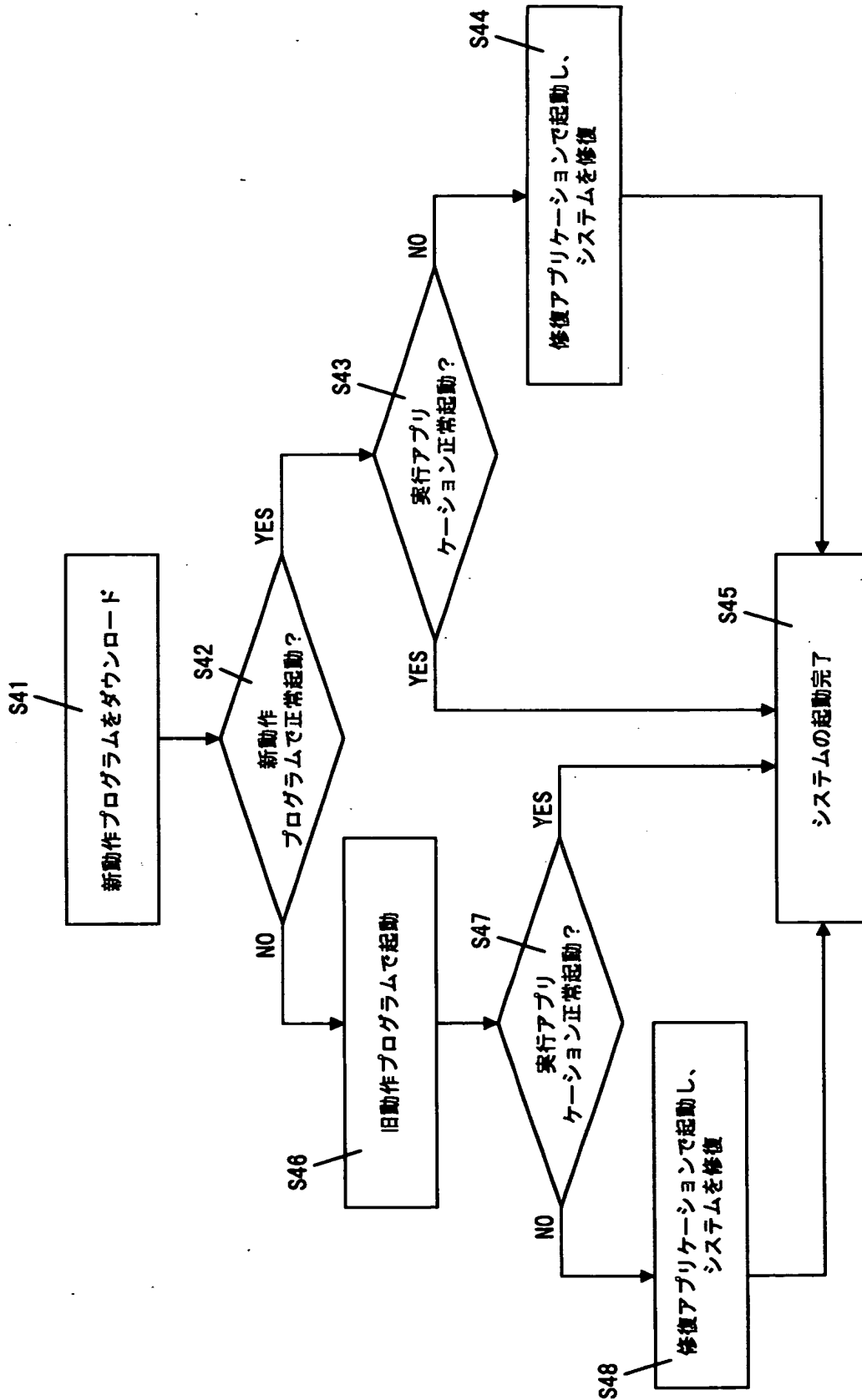
【図 13】



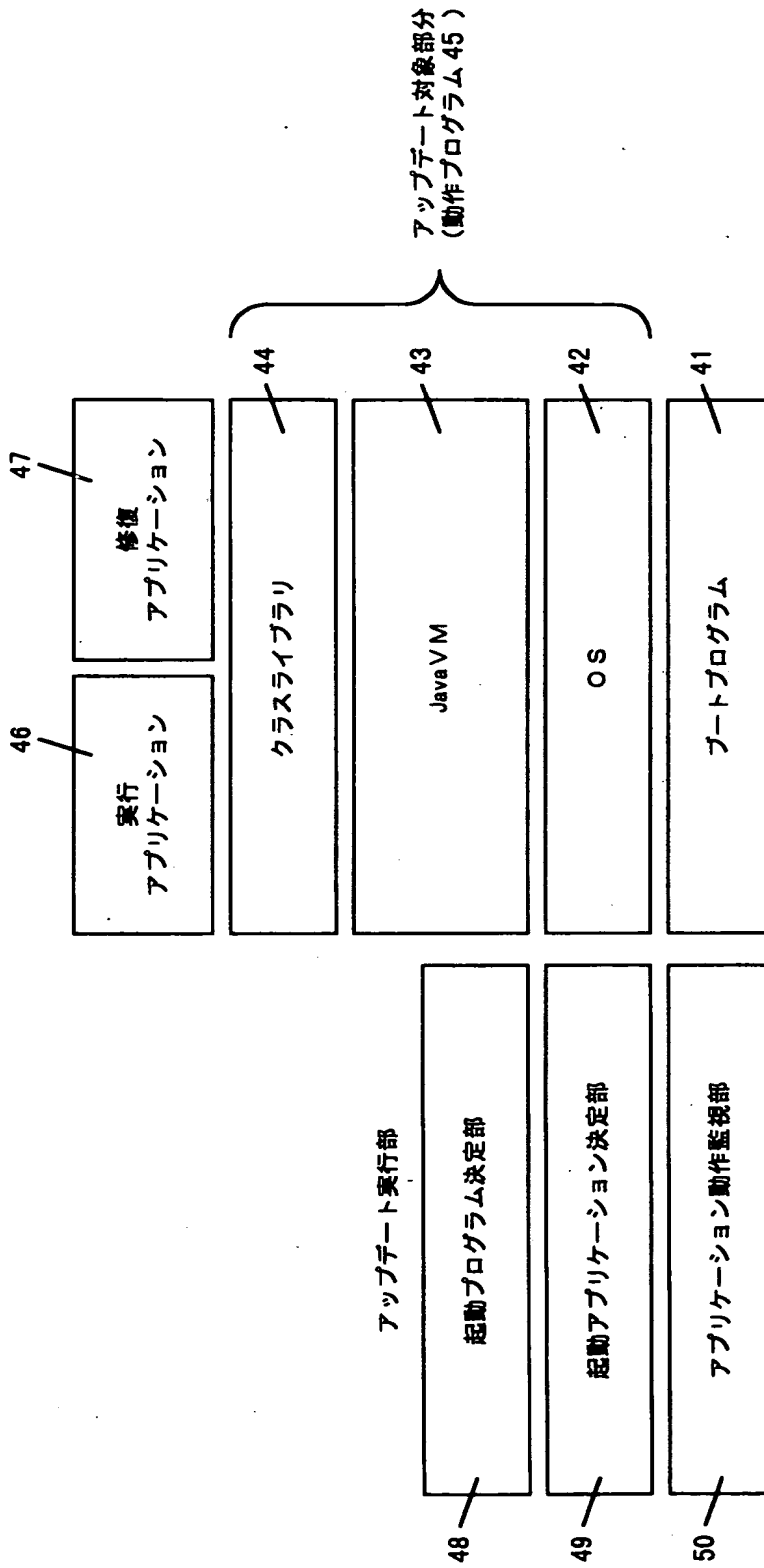
【図 14】



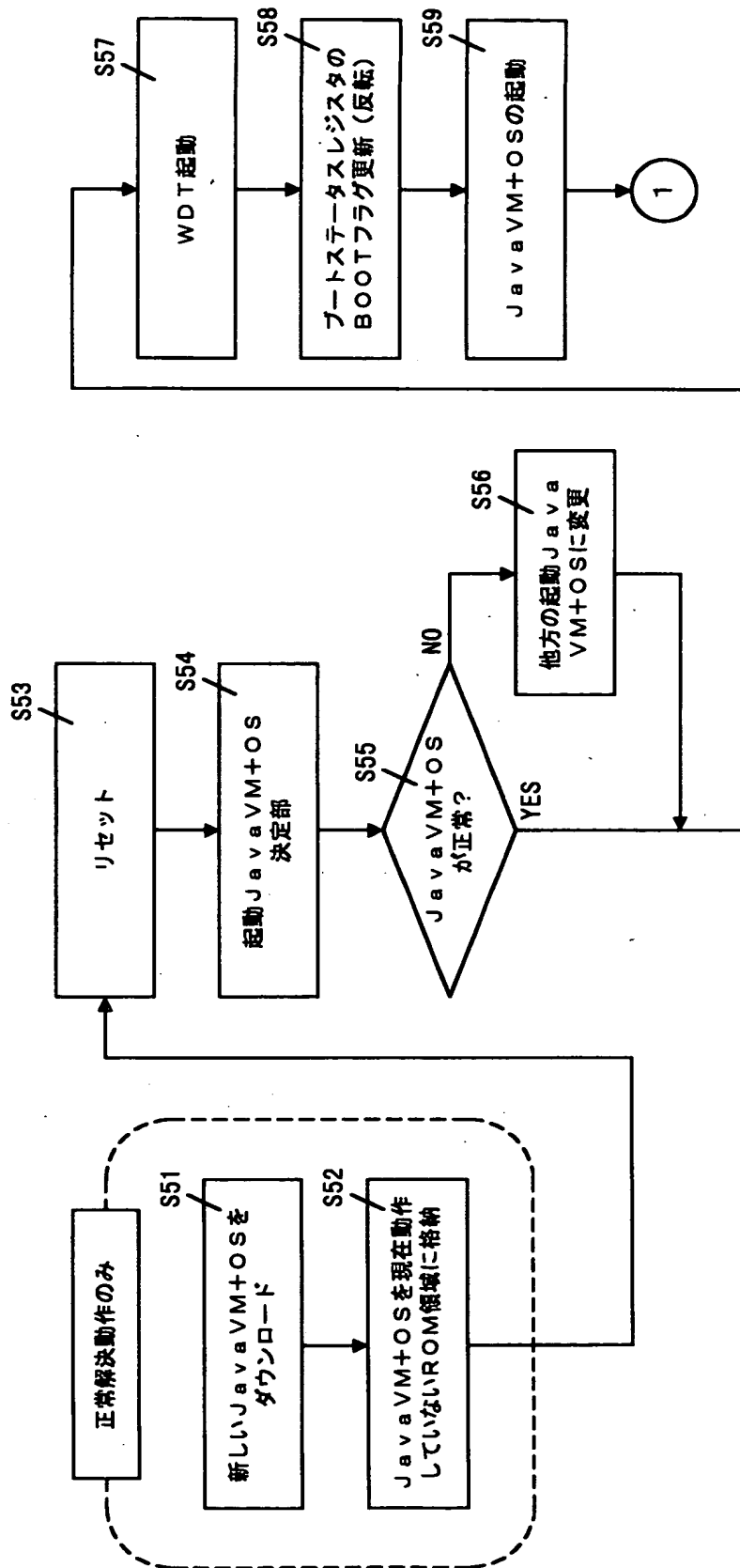
【図 15】



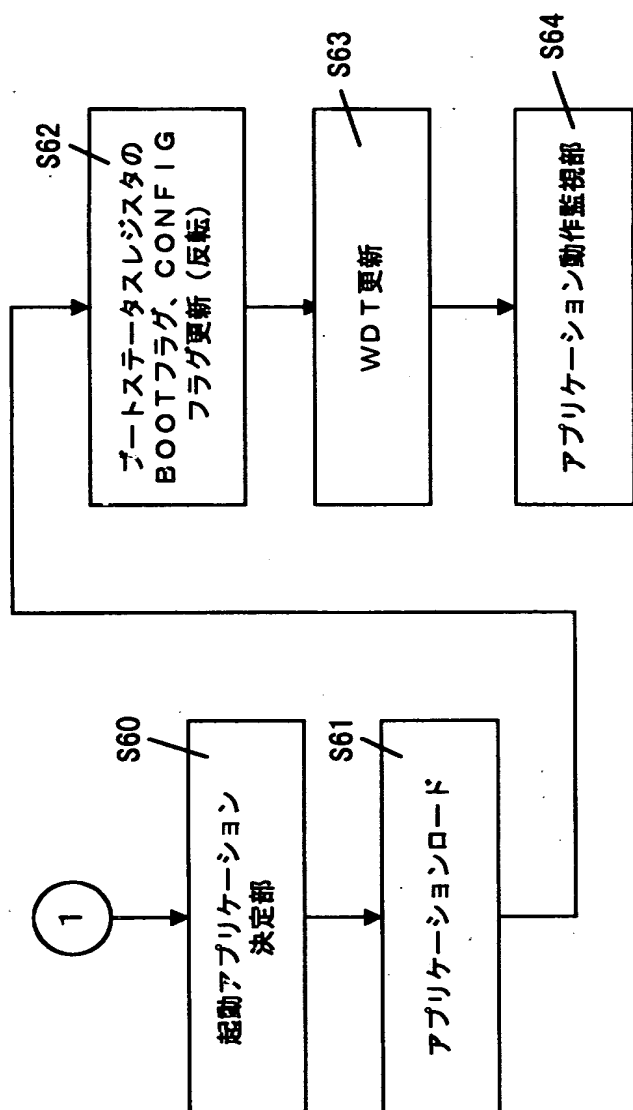
【図 16】



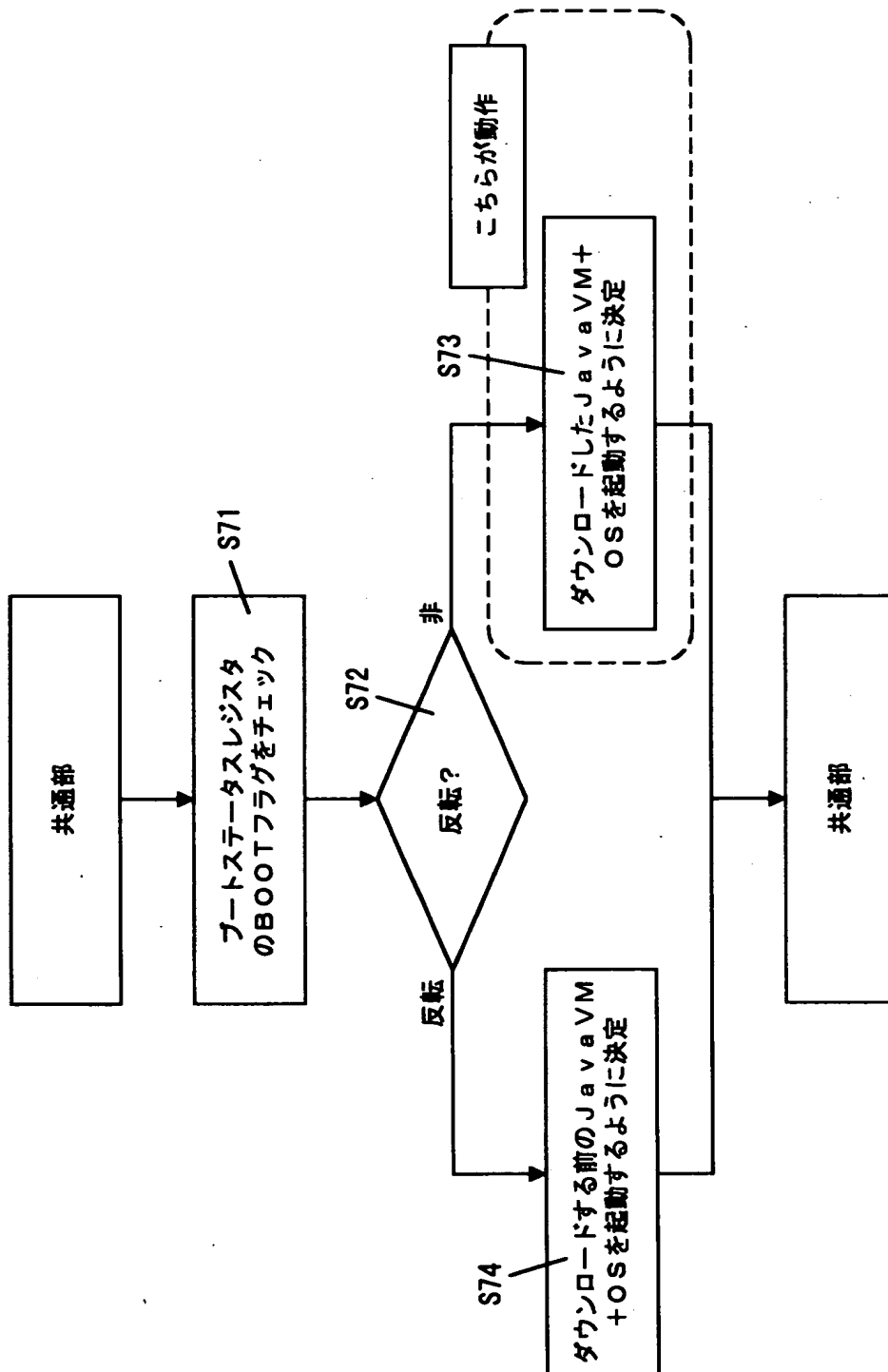
【図 17】



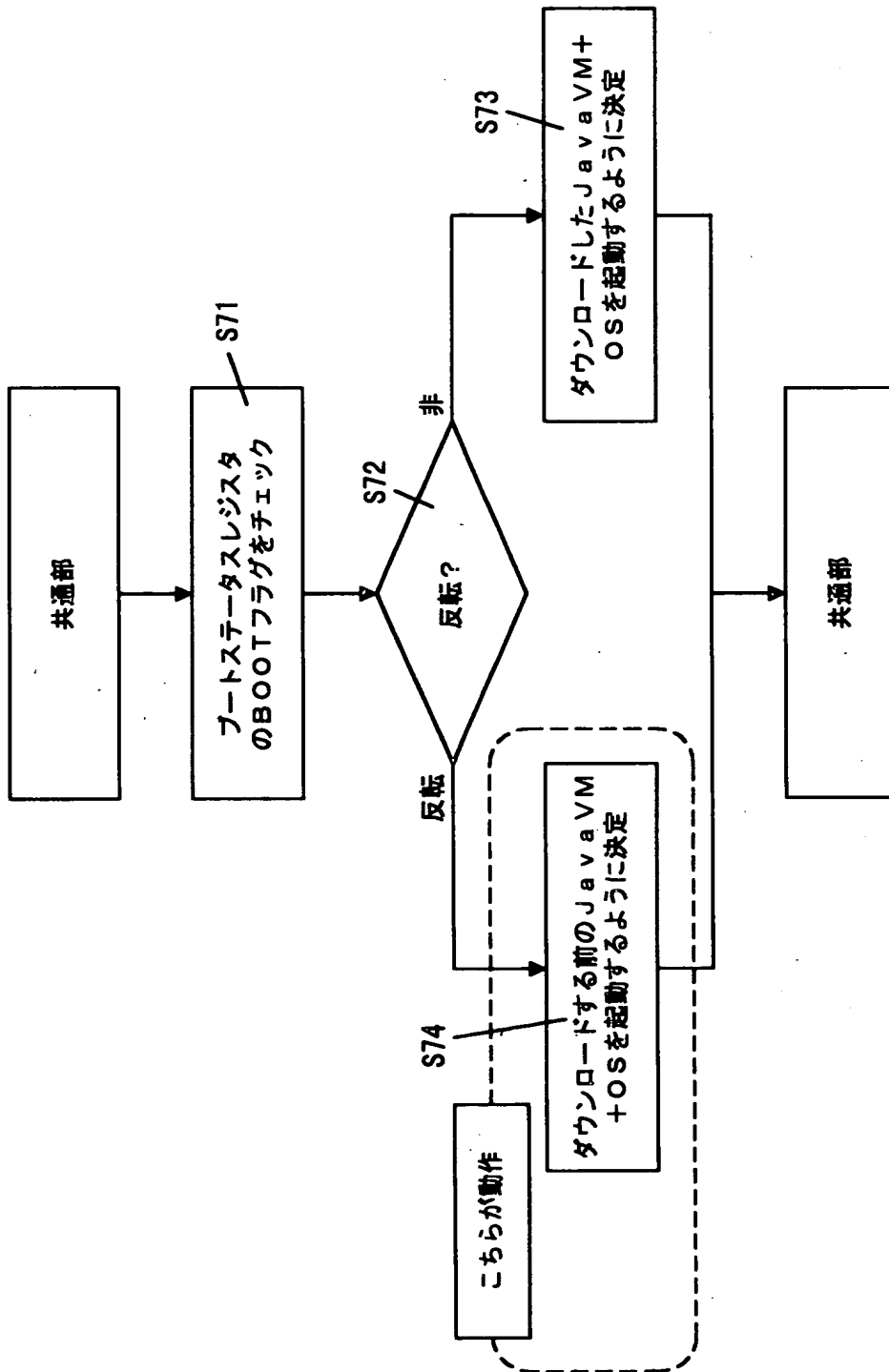
【図 18】



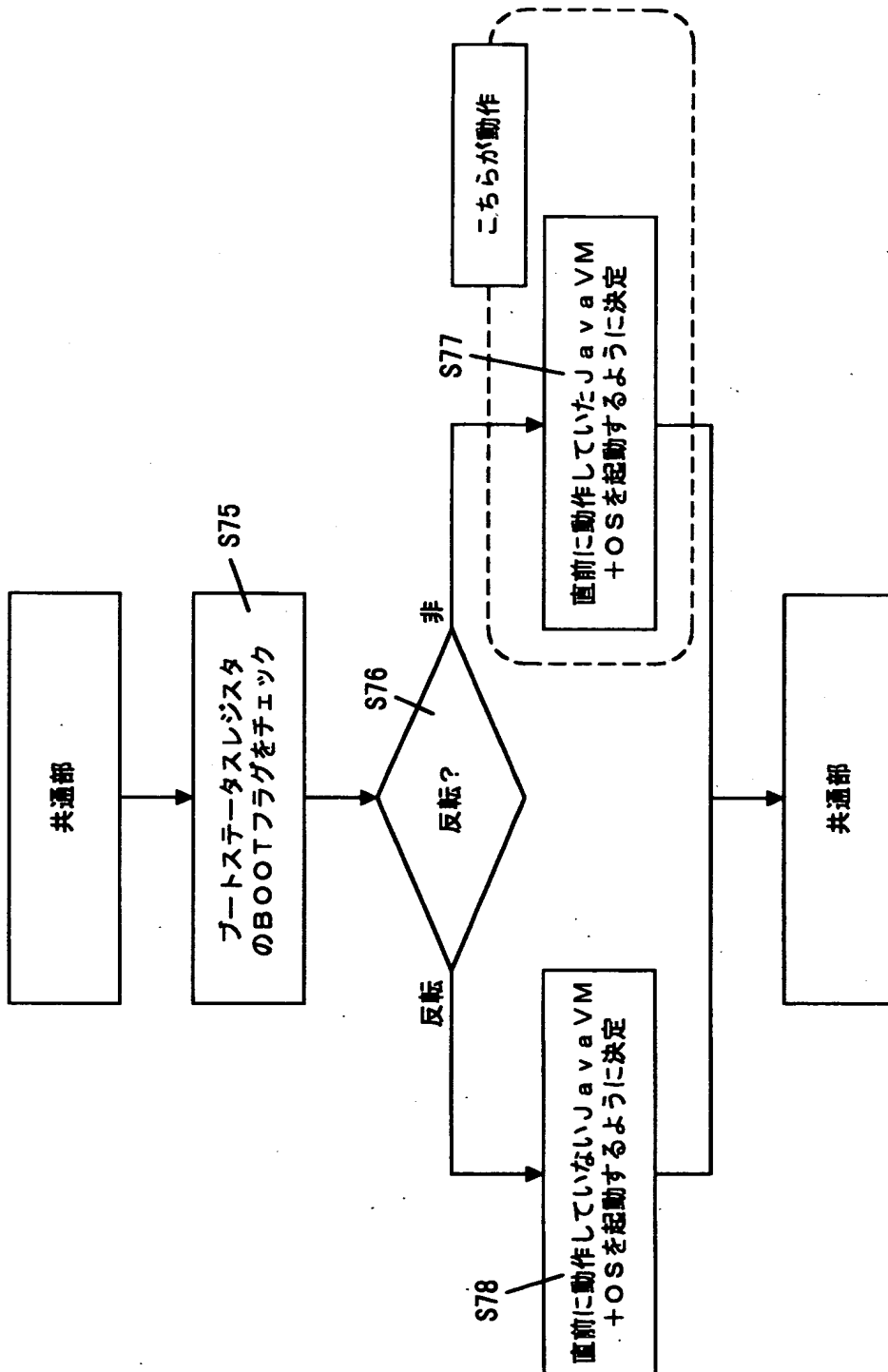
【図19】



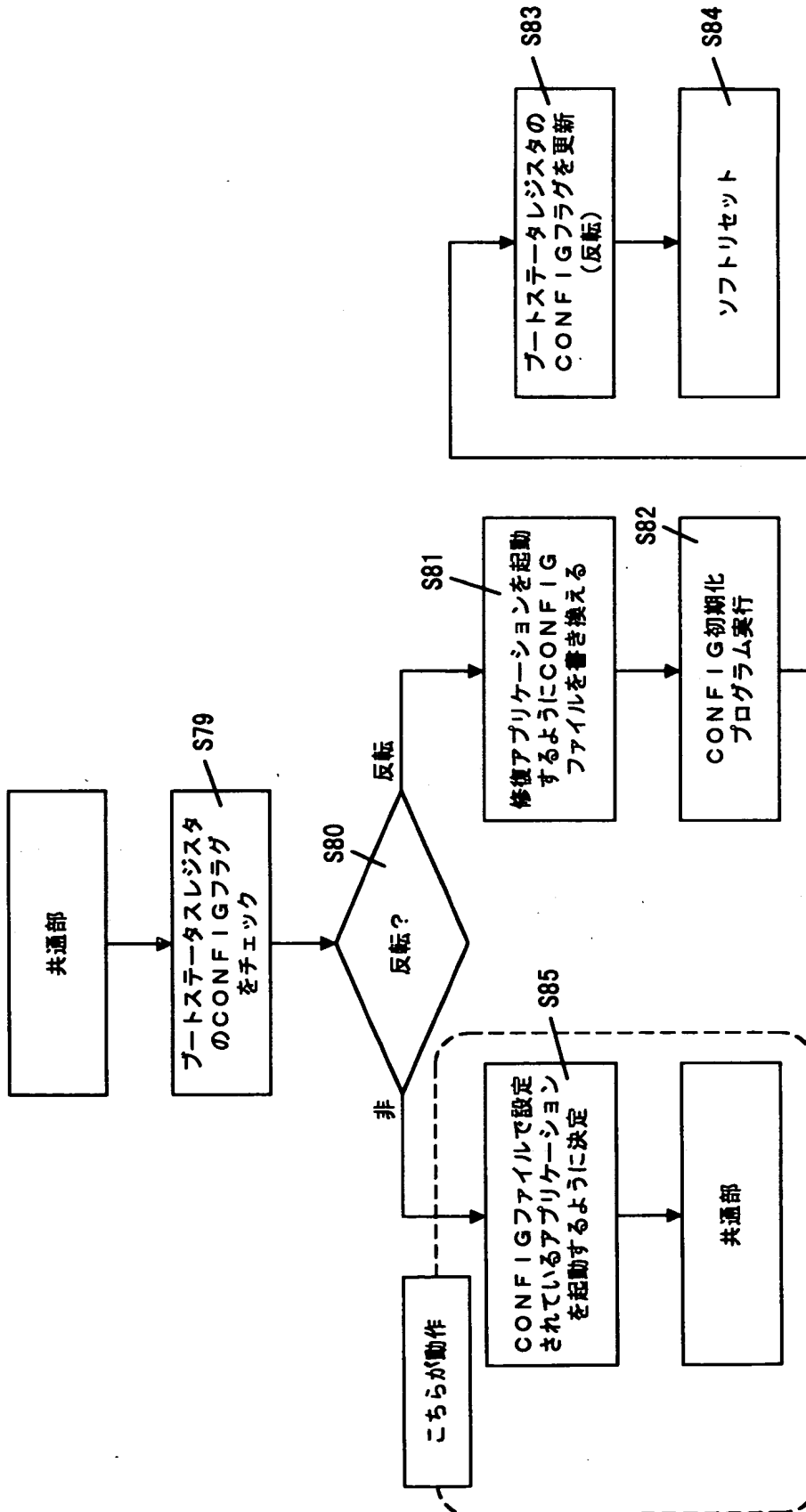
【図 20】



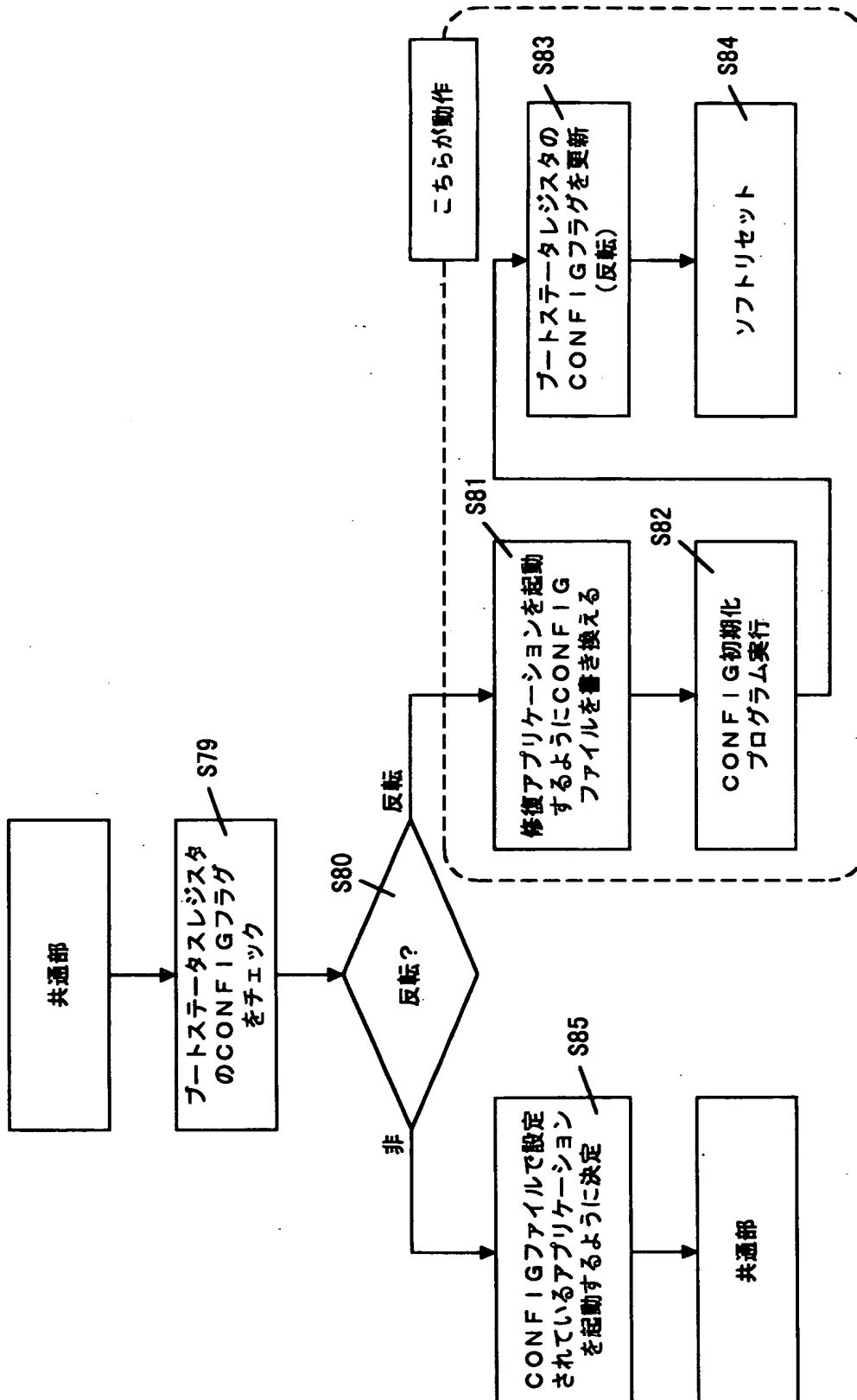
【図21】



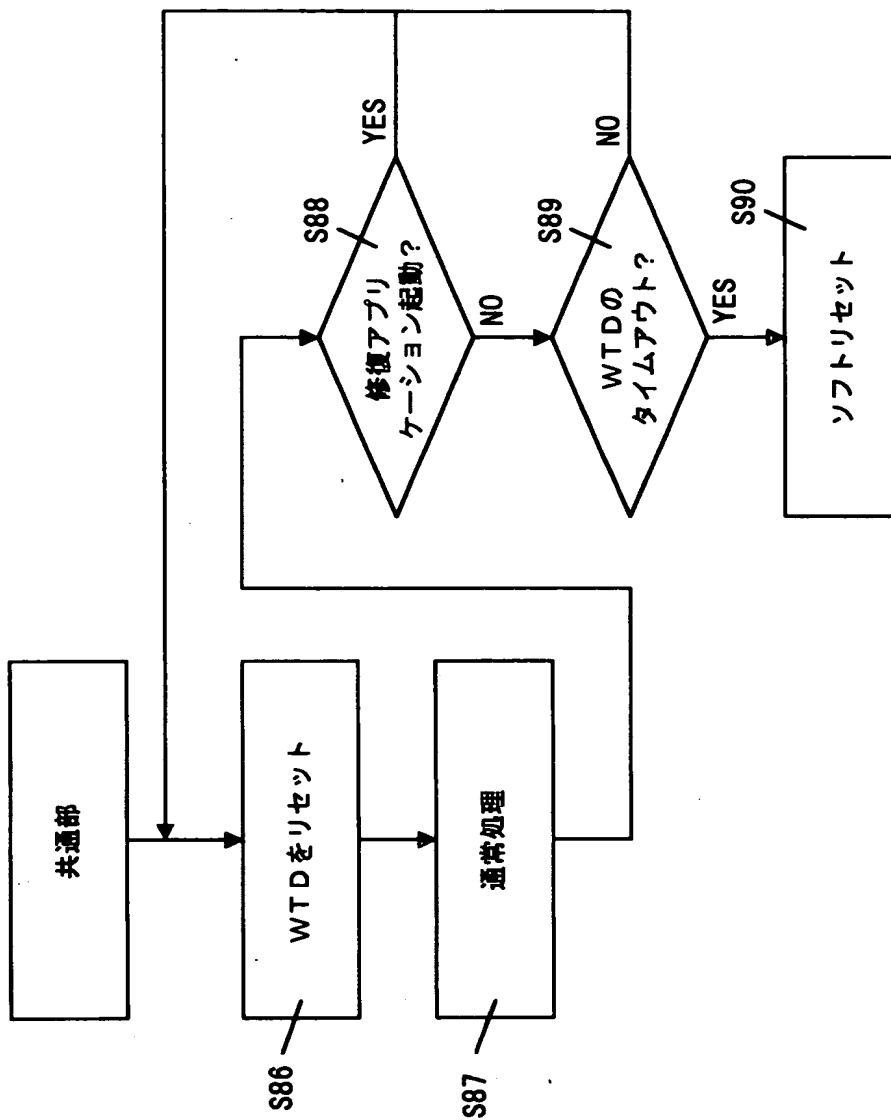
【図22】



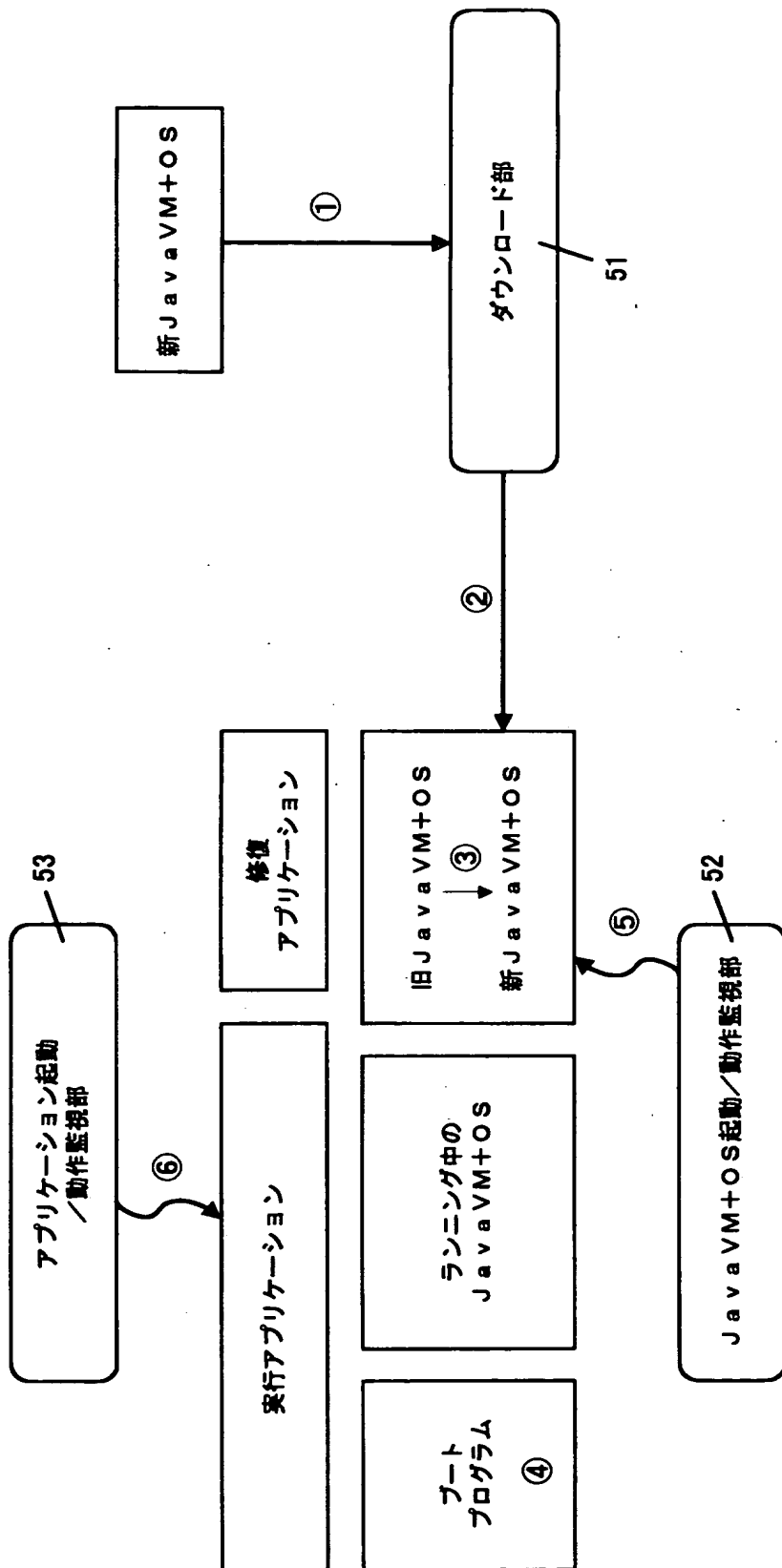
【図 23】



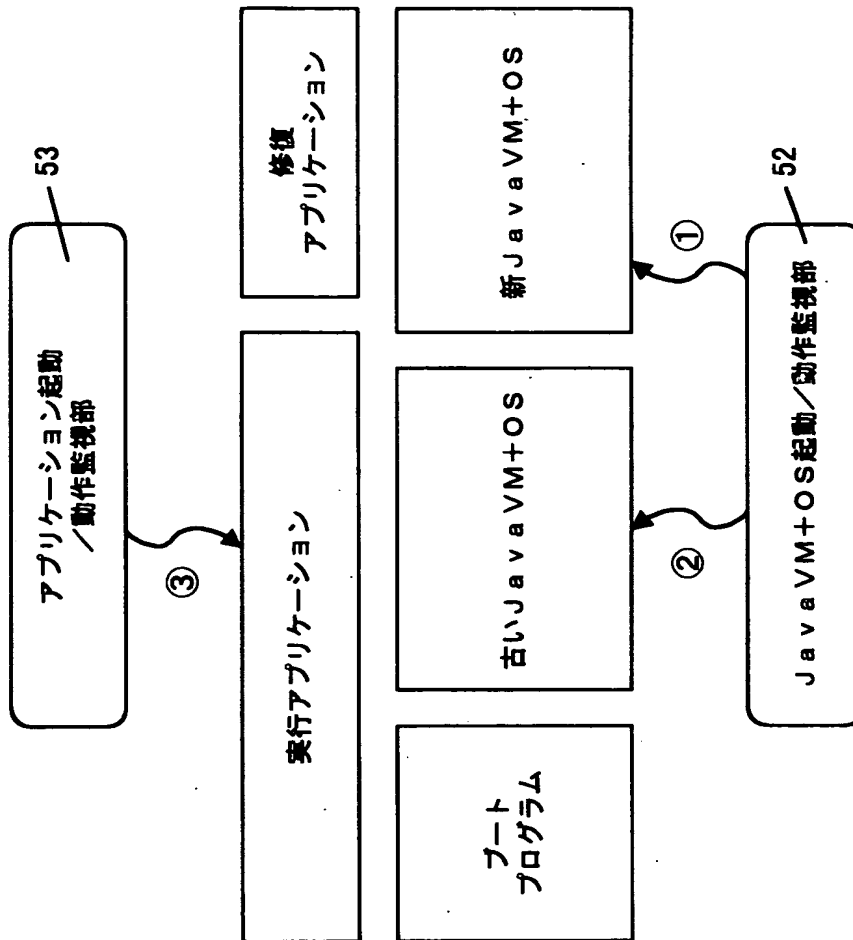
【図 24】



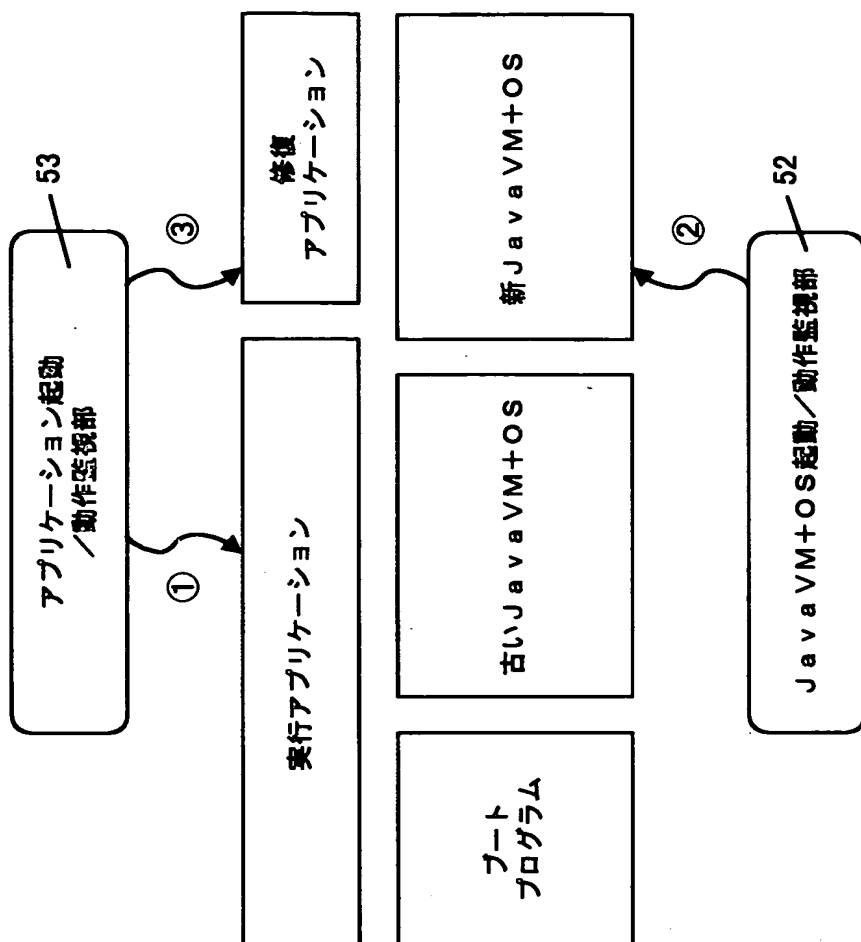
【図 25】



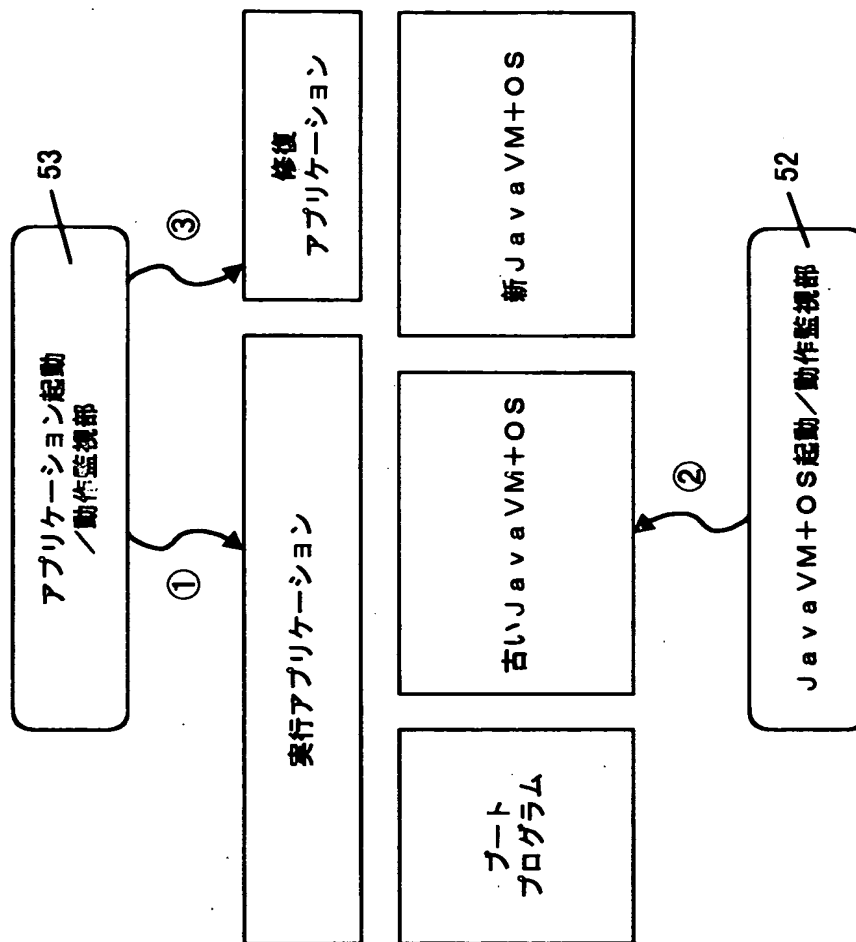
【図 26】



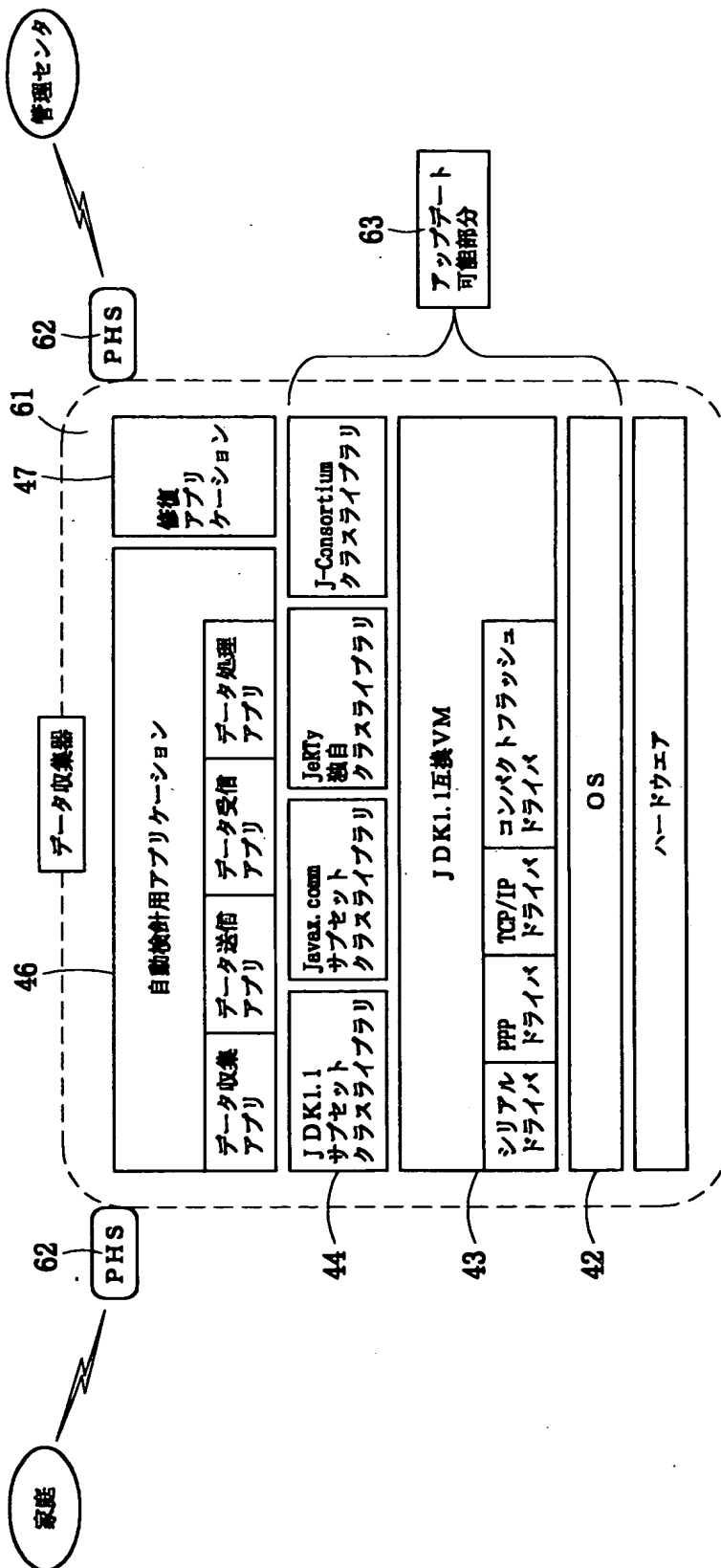
【図 27】



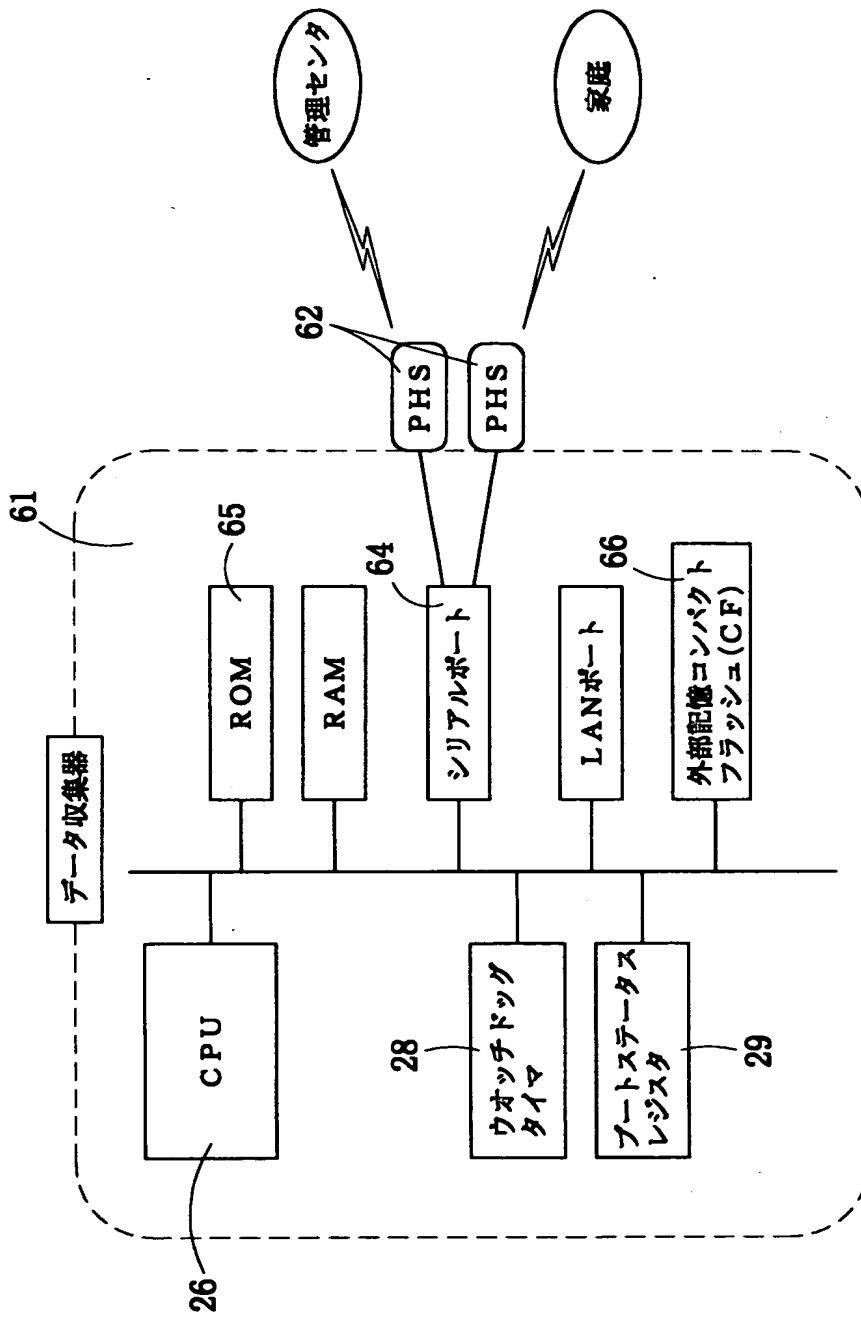
【図 28】



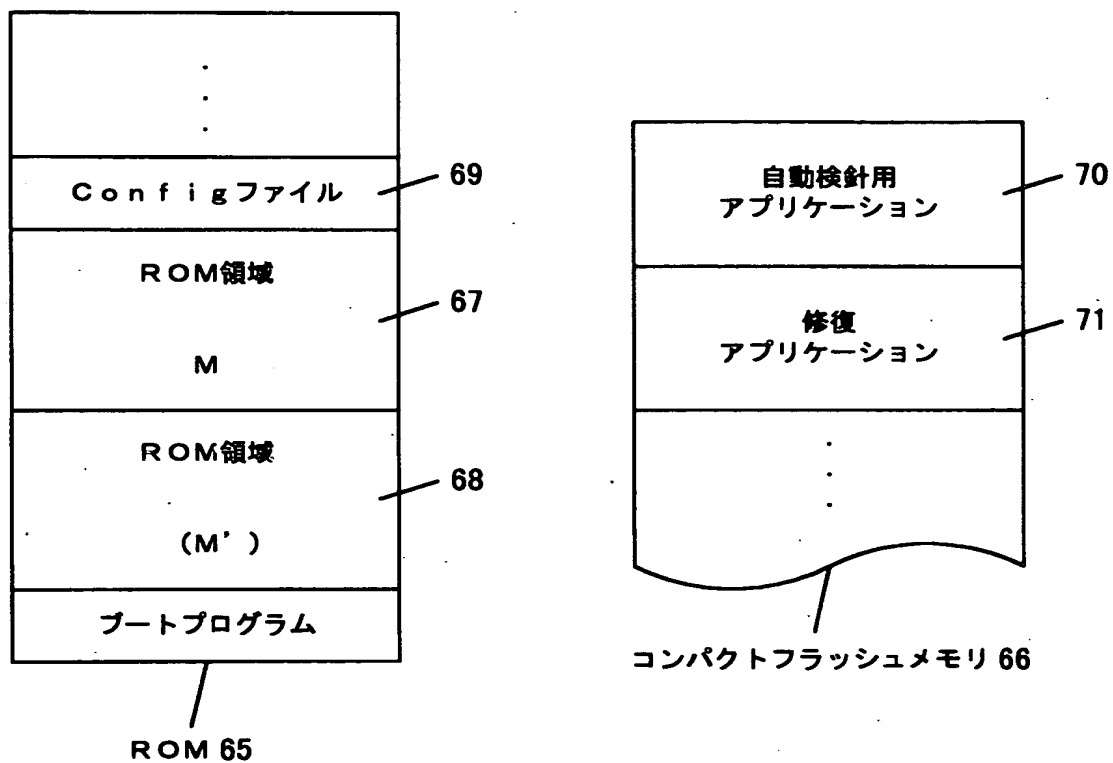
【図 29】



【図 30】



【図 3 1】



【書類名】 要約書

【要約】

【課題】 機器組込みシステム（特に、常時動作する必要がある機器のシステム）において、自動的にシステムのアップデートを行え、しかもシステムをフリーズさせることのないプログラム更新装置を提供する。

【解決手段】 従来動作していた旧動作プログラム 2 1 とは別な記憶領域に、ダウンロードした新動作プログラム 2 4 を格納する。ついで、新動作プログラム 2 4 を起動させ、新動作プログラム 2 4 の上で実行アプリケーション 2 2 を動かす。新動作プログラム 2 4 が起動に失敗した場合には、旧動作プログラム 2 1 を再起動して旧動作プログラム 2 1 でシステムを動かす。また、新動作プログラム 2 4 又は旧動作プログラム 2 1 が起動に成功した状態で、実行アプリケーション 2 2 が起動できなかった場合には、修復アプリケーション 2 3 によって実行アプリケーション 2 2 を修復する。

【選択図】 図 8

出 願 人 履 歴 情 報

識別番号

[000002945]

1. 変更年月日 2000年 8月11日

[変更理由]

住所変更

住 所

京都市下京区塩小路通堀川東入南不動堂町801番地

氏 名

オムロン株式会社



Creation date: 05-18-2004
Indexing Officer: KCOOPER4 - KEISHA COOPER
Team: OIPEBackFileIndexing
Dossier: 10091369

Legal Date: 08-22-2002

No.	Doccode	Number of pages
1	IMIS	1

Total number of pages: 1

Remarks:

Order of re-scan issued on